

【事前準備のお願い】中級(2/3)

ワークショップ#3では複雑な会話を目指します。

配布資料

`ws3_textfile_forwindows` / `ws3_textfile_formac`

もしくは

<https://goo.gl/YfEKac>

をご準備いただけると

効率よく開発ができます。

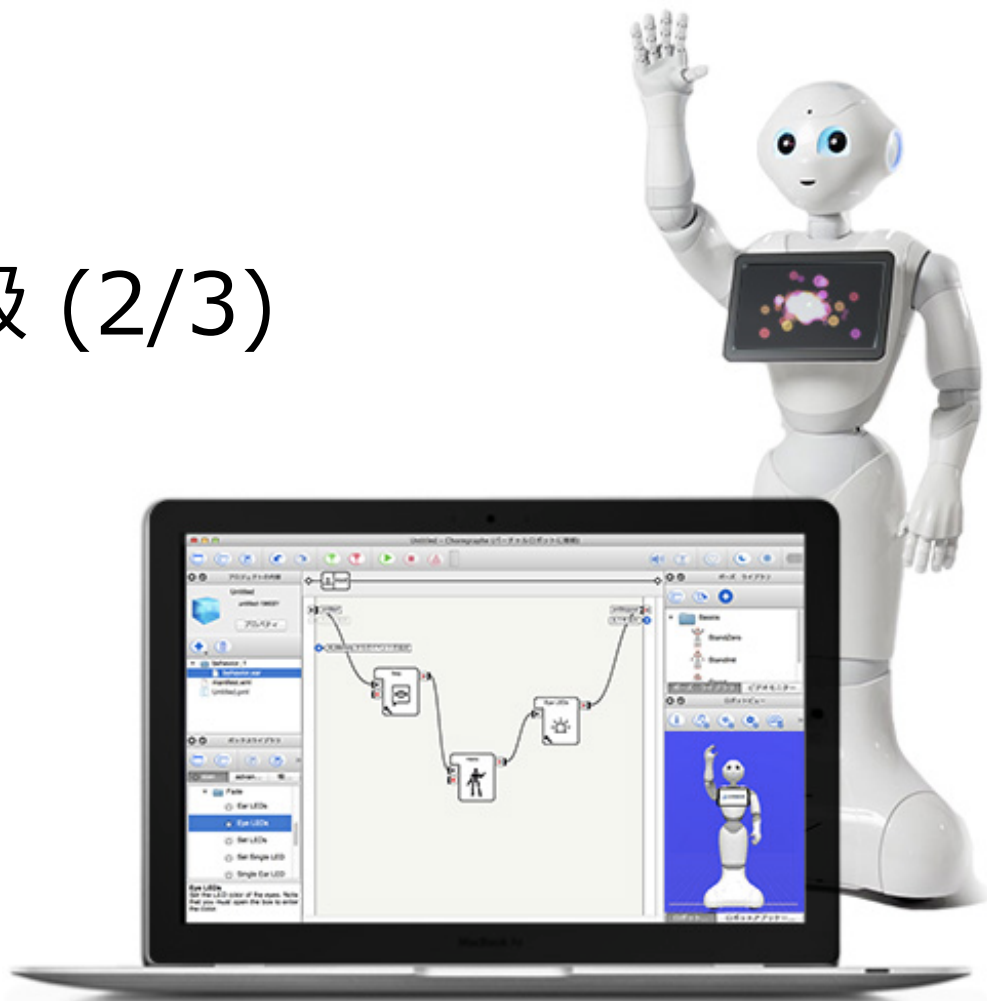


Atelier Akihabara

ワークショップ中級 (2/3)

2017/5/21

Softbank Robotics



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

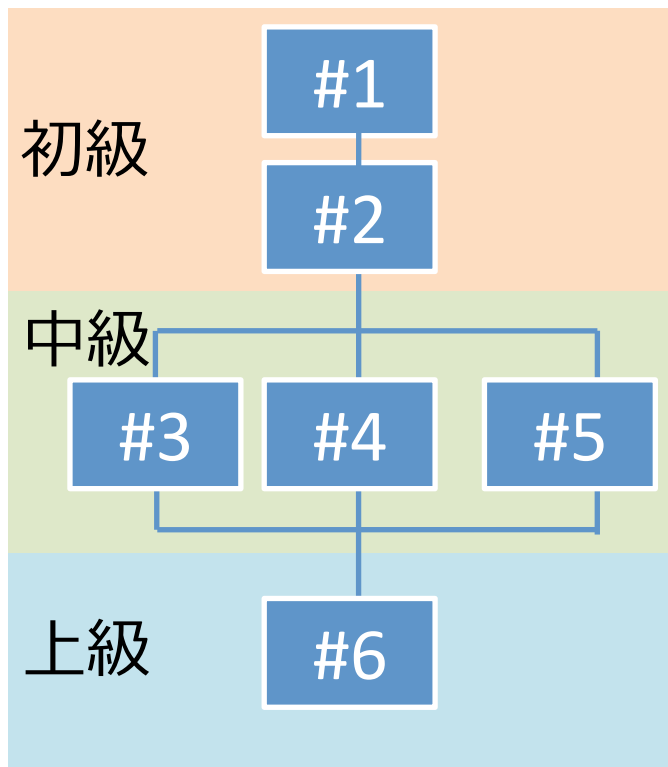


✓ 経験や知見を共有

相互
促進

アトリエについて

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

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の中身を知ろう
- ・ 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(関数を作る)
- ・ Pepperで学ぶPython基礎講座その3(BOXを編集)
- ・ 既存のBOXをPythonで書きかえてみよう(メールとQRコード)

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



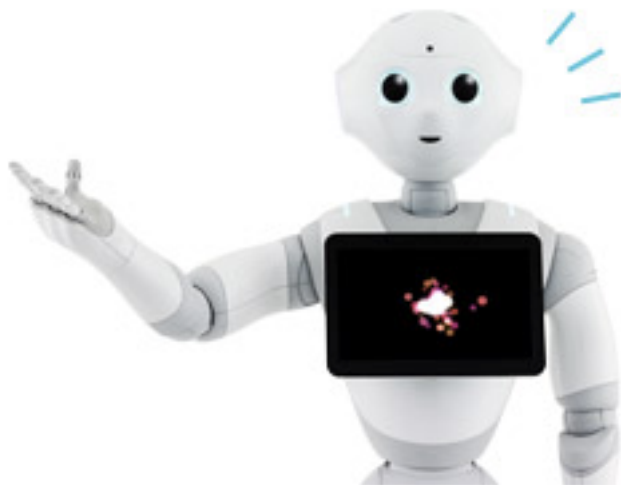
- ・ **お名前**
- ・ **所属**
- ・ **本日の意気込み**

例：

本日の案内を勤めさせていただきます、
***と申します。

- 1.qichatとは
- 2.qichatを使った会話
- 3.他のボックスとの連携

Qichatとは？



①簡単なスクリプトで多様な表現ができる

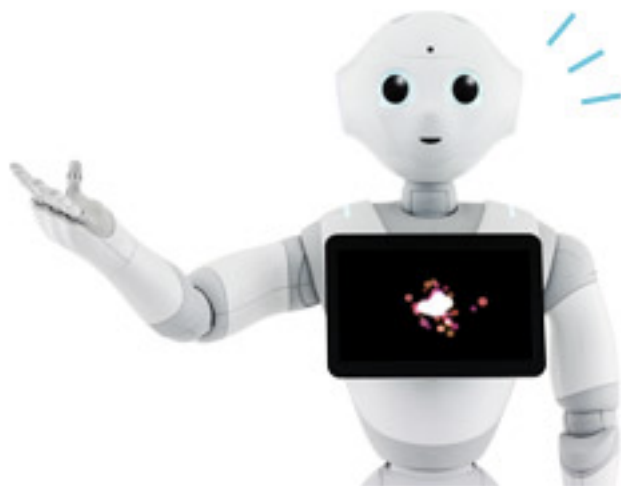
- 人間との応対に特化したスクリプト
- Animated Sayで使った動作や他のアプリを動かすことができる

台本を書くようなイメージでPepperの会話をつくることができる。

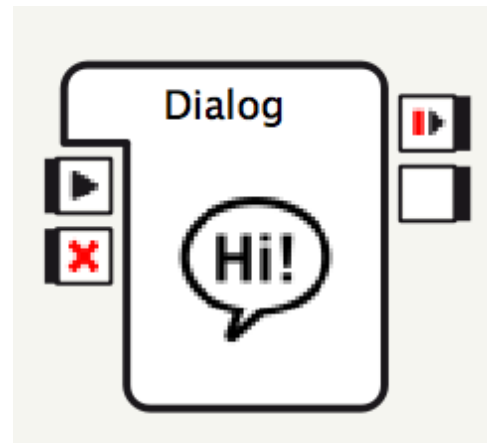
②人間らしい入力に対応

- 言いよどみや言い換え、語順の変化などにも対応

Speech Reco.ボックス（キーワード認識）では難しかった
「人間らしい会話」をさせることができる。

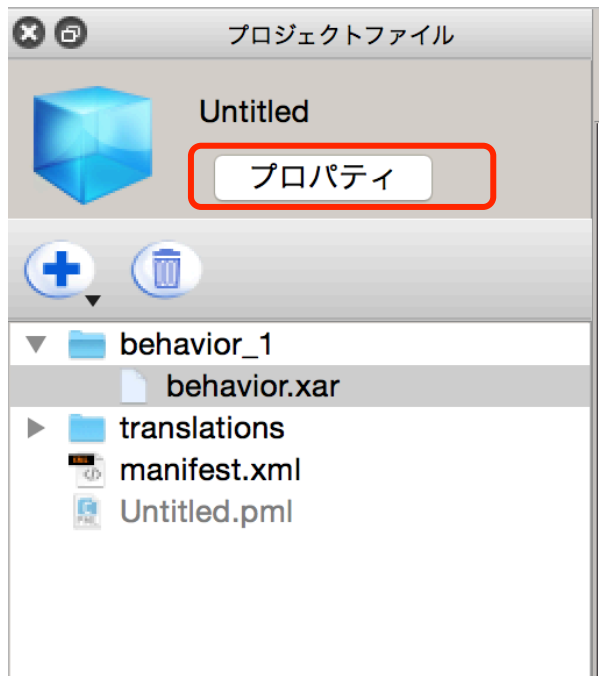


使うボックスはこれ

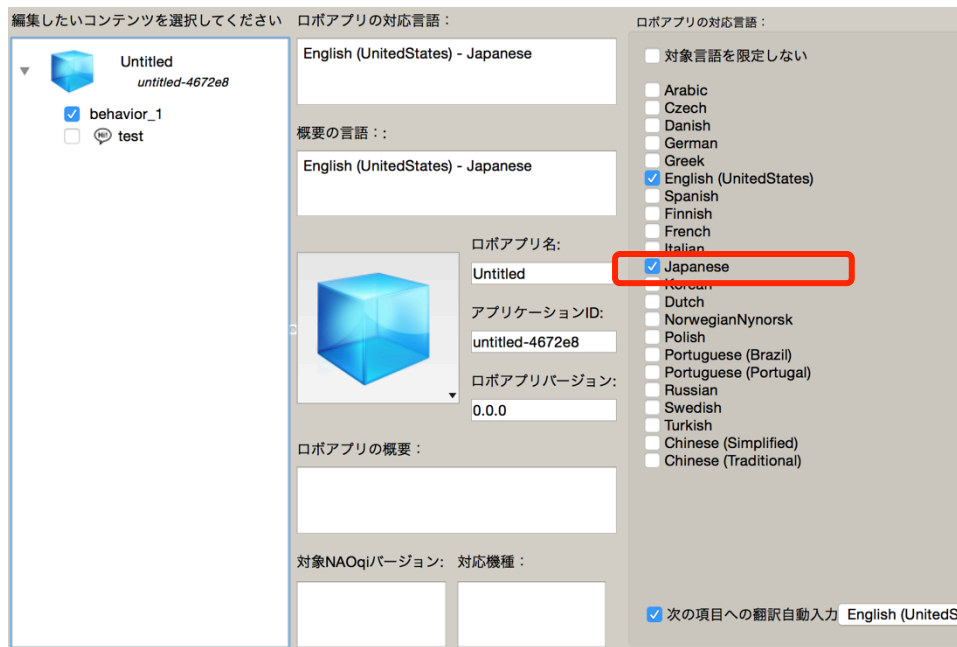


ダイアログトピックの作り方①

①プロジェクトファイルのプロパティをクリック

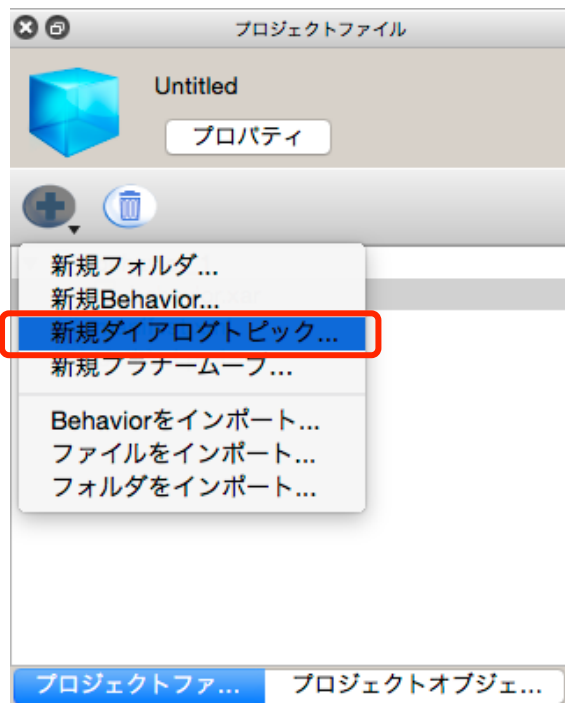


②Japaneseにチェック



ダイアログトピックの作り方②

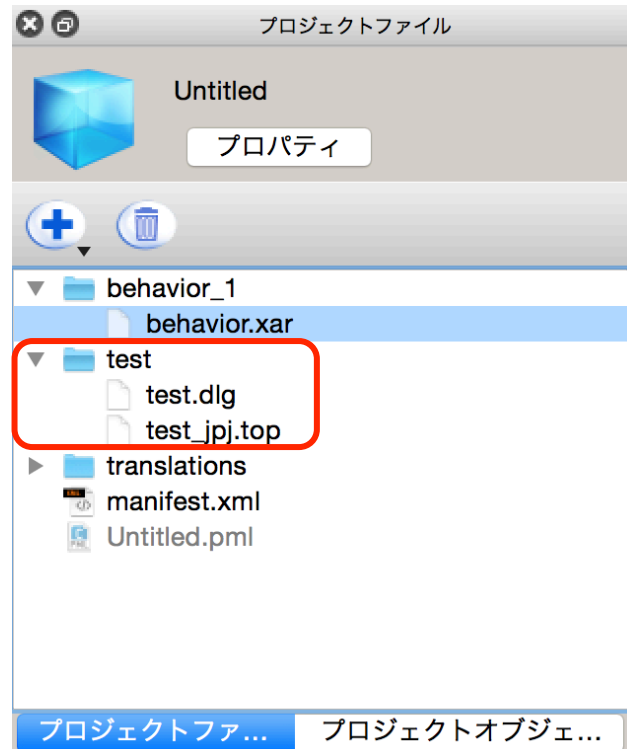
③新規ダイアログトピック
を選択



④名前を設定し、
日本語にチェック

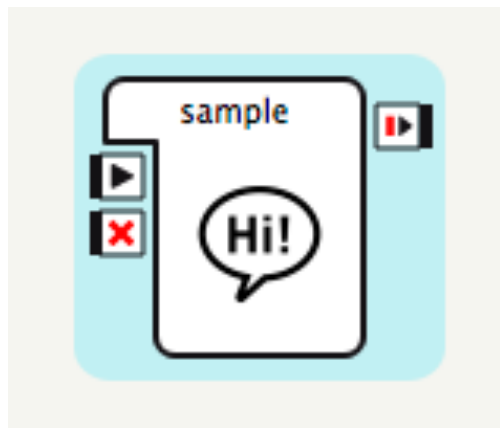


⑤トピックファイルが生成



ダイアログトピックの作り方②

①自分の設定したトピック名のフォルダから*.dlgファイルをドラッグ&ドロップすると、ボックス化され、他のボックスと同じようにリンクでつなぐことができるようになる。

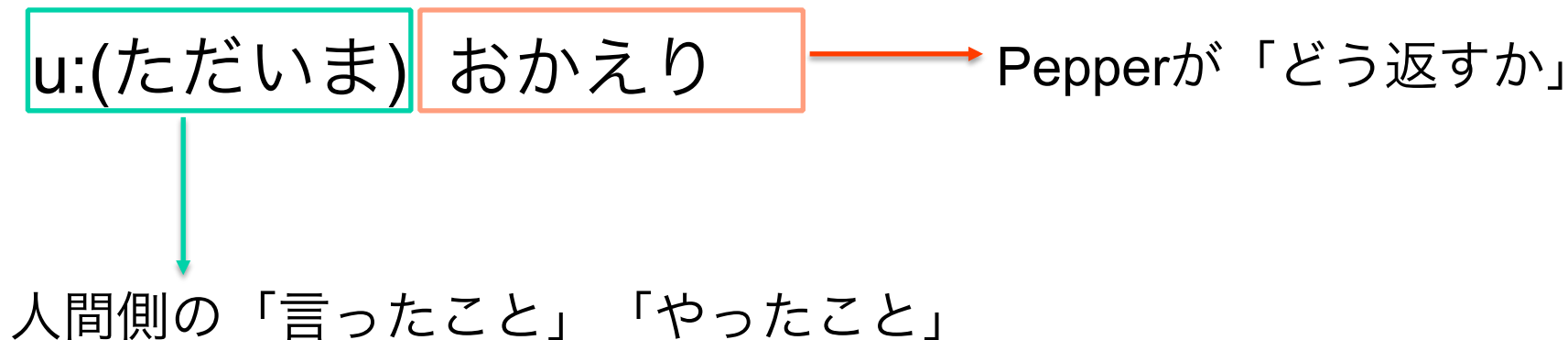


②*_jpj.topファイルをダブルクリックすると、会話の内容を設定できるようになる。

```
1  topic: ~sample()
2  language: jpj
3
4  # Defining extra concepts out of words or group of words
5  #concept:(hello) [hello hi hey "good morning" greetings]
6
7  # Catching inputs and triggering outputs
8  #u:(e:onStart) $onStopped=1
9
10 # Replying to speech
11 #u:(~hello) ~hello
12 |
```

— 1 文目 — 基本のカタチ

User rule

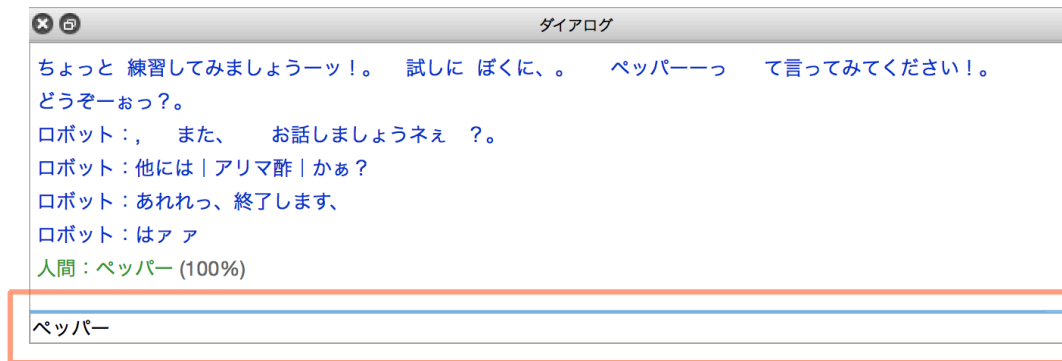


※カッコの半角、全角、半角スペースの挿入などに注意。

ダイアログのコツ

1.真正面、30cmから話す

2.ダイアログビューを使う



3.ダイアログ機能の閾値を変更する

PepperのDialogの閾値変更Boxを作ってみた

<https://goo.gl/bwdjEO>

TerminalからqicliコマンドでPepperのDialogボックスの認識のしきい値を変更する

<https://goo.gl/GEblCC>

参考： タグによる調整

パラメータ	意味	タグ
Voice Shaping	声のピッチ（高低）を決めるパラメータ	\vct=（50～200の数値）\
Speed	声の速度を決めるパラメータ	\rspd=（50～400の数値）\
Pause	待機時間 ミリ秒単位で設定	\pau=（数値）\
Reset	すべての設定値を初期設定に	\rst\

\rspd=110\vct=135\ に設定するとPepperらしい澆刺とした声になります。

“\rspd=75\vct=135\ ふっ \vct=110\ ふっ \vct=100\ ふっ。”
などオートノマスライフの会話でも使われています。

—2文目— u1.u2...を会話の流れを設定する

User subrule

u: (動物の話) 犬か猫を飼っていますか？

u1: (犬) 大きい犬ですか？

u2: (はい) 広い場所が必要ですね

u2: (いいえ) かわいらしいですね

u1:(猫) 田舎に住んでいますか？

u2:(はい) 外に出ていきますか？

u3:(はい) 鼠を捕ってきますか？

u2:(いいえ) とても大きな家にお住みなんですね

u1:(いいえ) 私と一緒にですね

Choice

u:([こんにちは おはよう こんばんは])

[こんにちは おはよう こんばんは] 今日頑張ろう！

→[]と半角スペースで
ORと同じ意味合い

—3文目— 曖昧な表現に対応する OR

^randを[]と半角スペースの前に入れることで
ランダム再生ができる

```
#ただいま
```

```
concept:(greetings_comein_high)^rand[
```

```
"[^start(sbr_90070_motion_pepper_talk/y_greet-anime-hi)],\pau=700\\vct=155\\rspd=110\御帰りいイツ！\pau=2000\ "  
"[^start(sbr_90070_motion_pepper_talk/y_greet-anime-katate-gatsu)],\pau=1000\\vct=150\\rspd=110\御帰りイイツツツツツ  
"[^start(sbr_90070_motion_pepper_talk/y_greet-anime-hiroge-desuka)],\pau=600\\vct=160\\rspd=90\おっか襟いいいっ？。  
"[^start(sbr_90070_motion_pepper_talk/y_greet-anime-ossu)],\pau=1200\\vct=155\\rspd=135\おかえりーいっ\pau=1000\  
"[^start(sbr_90070_motion_pepper_talk/y_cocoro-anime-perfect)],\pau=800\\vct=150\\rspd=115\おかえり\pau=165\なサーー  
"[^start(sbr_90070_motion_pepper_talk/y_greet-anime-hand-pa)],\pau=300\\vct=150\\rspd=110\お力襟\pau=160\ナさあー  
"[^start(sbr_90070_motion_pepper_talk/y_greet-anime-katate-ojigi)],\pau=1100\\vct=160\\rspd=110\お返りなさいー！\pa  
"[^start(sbr_90070_motion_pepper_talk/y_greet-anime-hand-pa)]\pau=145\\rspd=140\アアッ？\pau=300\\vct=150\\rspd=115\  
"[^start(sbr_90070_motion_pepper_talk/y_greet-anime-ureshiii)],\pau=1200\\vct=145\\rspd=115\お力襟\pau=160\ナさー
```

オートノマスライフの中でも使われている

—4文目— コンセプトを設定する

Concept

concept:(yes) [はい うん そうだよ 元気だよ]
concept:(no) [いいえ 全然 調子悪い]

u:(こんにちは) こんにちは、お元気ですか

u1:(~yes) ボクも元気ですー！

u1:(~no) 大丈夫ですか？

→長い文に便利！

input_storing

```
u:(_[こんにちは おはよう こんにちは])
```

```
$1 今日も頑張ろう！
```

→_の後の入力を変数化して
\$1で呼び出すことができる

—6文目— 入力を変数にする

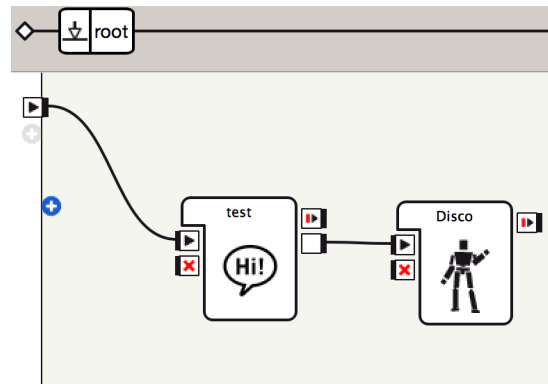
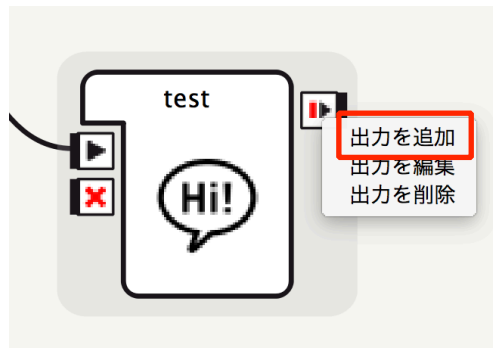
Variable

```
u:(_[チョコレート チーズ]がほしい) わかりました $askedFood=$1  
u:(なに頼んだっけ?) $askedFood  です
```

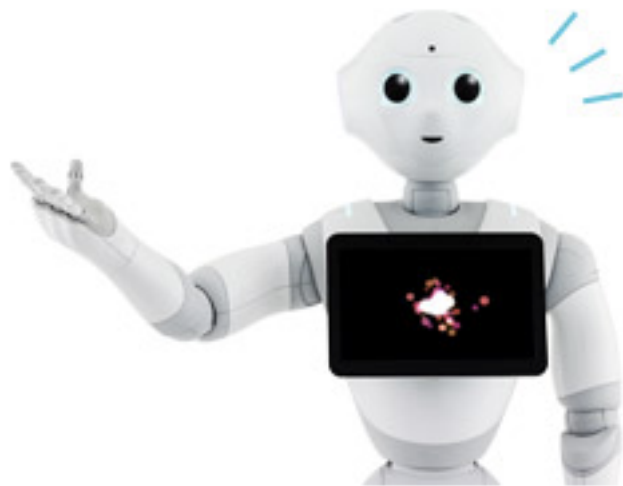
\$1に自由に名前をつけることができる

—7文目— 他のboxとの連動

u:(踊って) わかりました \$output=1



ダイアログボックスから
出力を増やすことができる



おまけ：
感情認識APIを
つかってみよう



笑顔認識アプリをつくってみよう

①新規にダイアログボックスを作成します。
(名前はわかりやすいように「smile」とします)



②以下の文章をダイアログのトピックファイルに記述します。

concept:(yes) [はい お願い そうだね やってみよう いいね]
concept:(no) [いいえ いいです またこんど]

u: (e:onStart) こんにちは。ところで笑顔チェックしませんか？

u1: (~yes)

^start(animations/Stand/Emotions/Positive/Happy_4)

オッケエ

^wait(animations/Stand/Emotions/Positive/Happy_4)

\vct=110\じゃあ僕に向かってえ \pau=1000\ \vct=170\ 笑顔
\vct=110\

\$checksmile=1

u1: (~no) そうですかあ。何か用事があったら呼んでね！

③ボックスを右クリックし、
[ボックスを編集]を選択

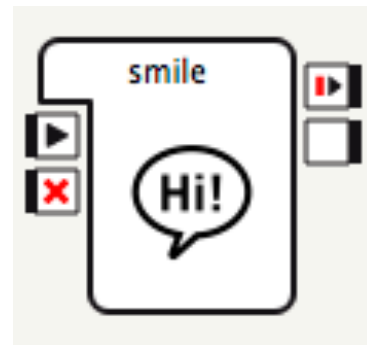


笑顔認識アプリをつくってみよう ③



④[A]出力の+ボタンをクリックし、[B]名前を「checksmile」としOKを選択

⑤smileボックスに新たな出力「checksmile」が追加されます。



Get Smileボックスの利用



笑顔が閾値を超えていたら出力

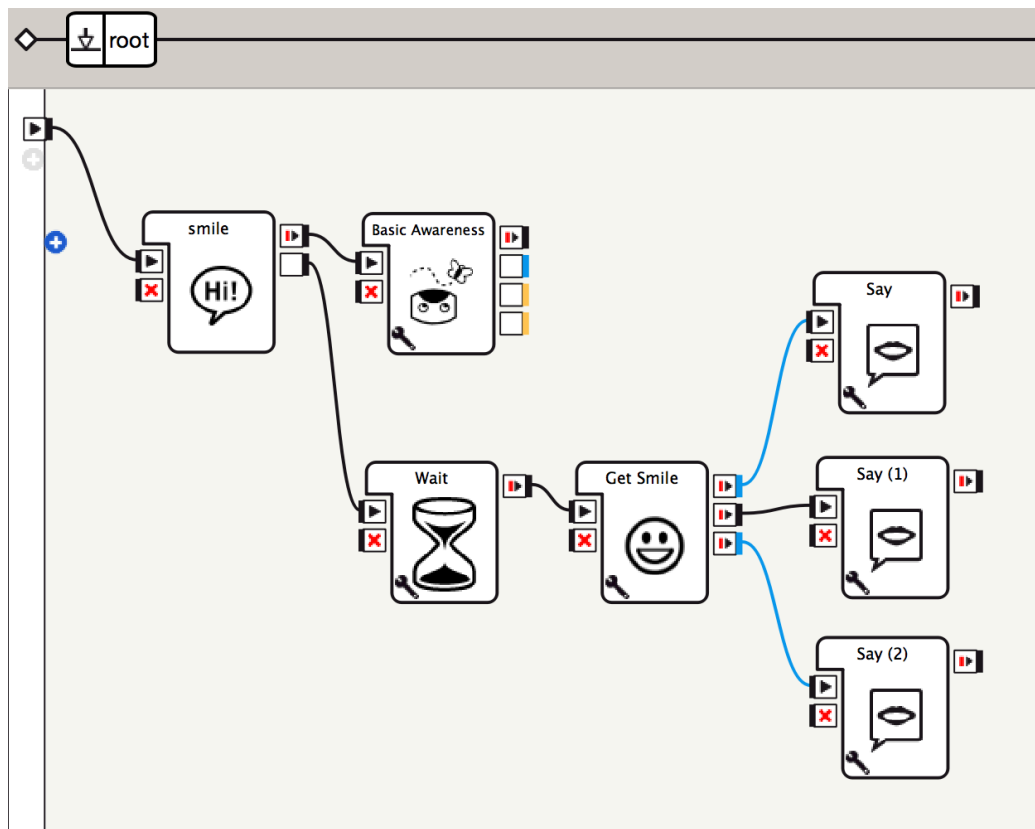
- little smile
 - middle smile
 - big smile
- の三段階で評価

笑顔が閾値を下回っていたら出力

エラー出力

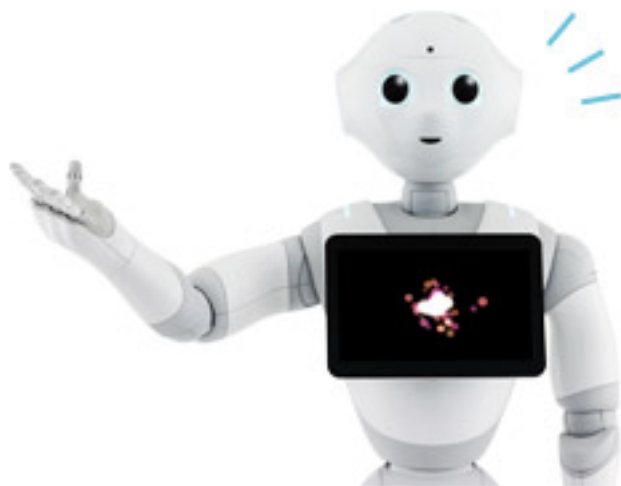
- No face detected
 - Multiple faces detected
 - Timeout
- の三種類

笑顔認識アプリをつくってみよう ⑤



⑥左のようにボックスを配置します。

⑦Waitボックスのタイムアウト、Sayボックスのテキストを変更し、完成です。



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

<https://goo.gl/ocUL9G>

- ・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

アンケートにご協力ください

<https://goo.gl/XoNkfd>



**おつかれさまでした！
これにてWS中級(2/3)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

