

# ようこそアトリエ秋葉原へ

このスペースは写真OKです。

ぜひハッシュタグ#pepper\_3331をつけて呟いてください

以下の事を確認してください

- (atelier-guest3)wifiに接続
- Choregrapheのバージョン(2.5.5)確認



アトリエ秋葉原

# ワークショップ番外編 Pepperのディレクトリ構造を知ろう

2018  
アトリエ秋葉原



# 実体験とコミュニティで開発を促進する

アトリエ

コミュニティ



Pepperのアプリ開発という  
実体験

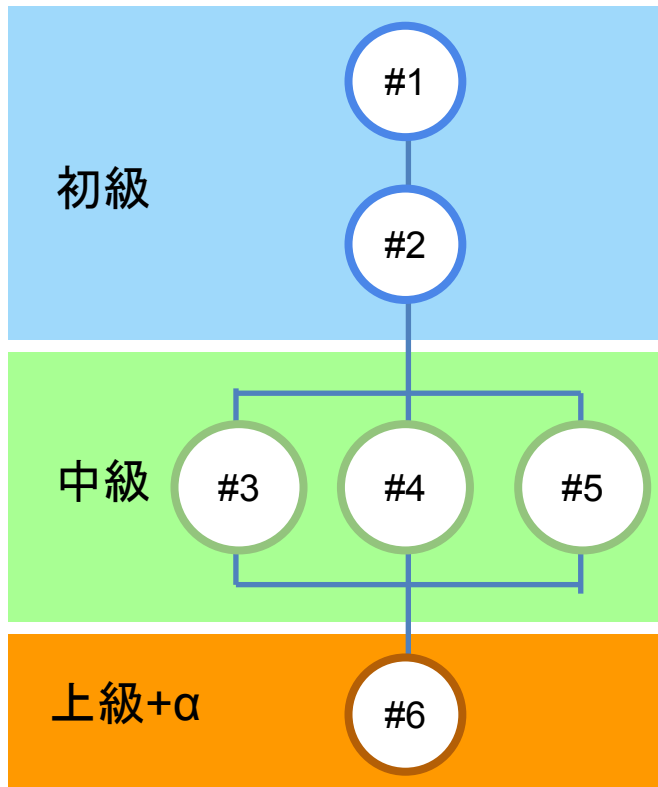
相互  
促進



経験や知見を  
コミュニティで共有

# アトリエ秋葉原について

## 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)
- ・既存のBOXをPythonで書きかえてみよう(メールとQRコード)
- ・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

## 軽く自己紹介をしましょう！

- お名前
- 所属
- プログラミング経験や本日の意気込み

今回ワークショップ講師を務める  
XXXXと申します。  
よろしくお願いします

## 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公式のものではないことを  
ご承知ください。



## 1. 写真を通じてPepperの中身に詳しくなる

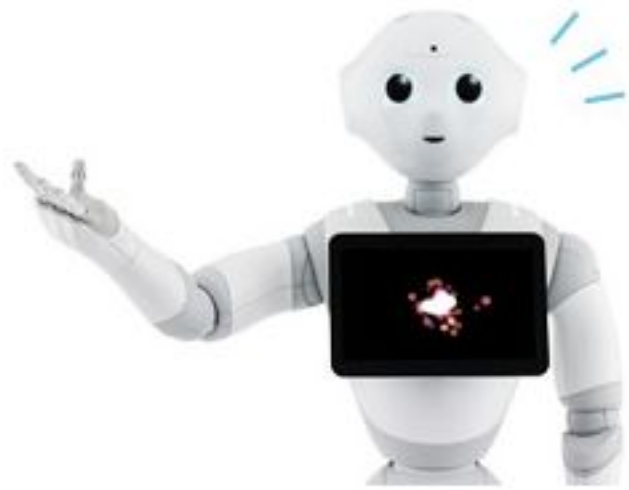
- a. 写真を撮る
- b. 写真をPCに転送
- c. 写真をタブレットに表示

## 2. SSHでPepperに接続

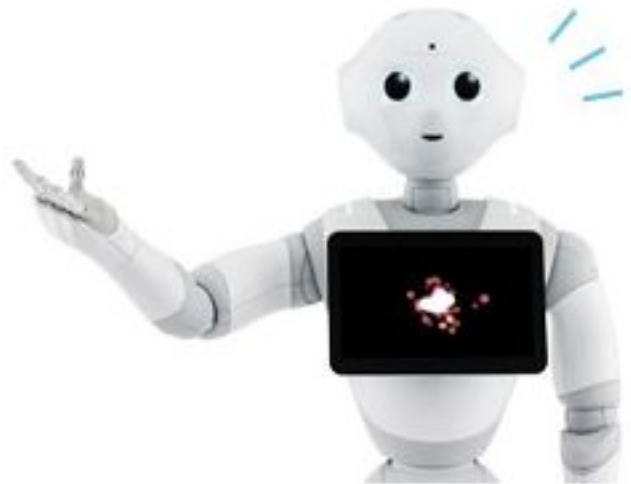
## 3. SSH接続でアプリ実行

## 4. Pepperログの確認

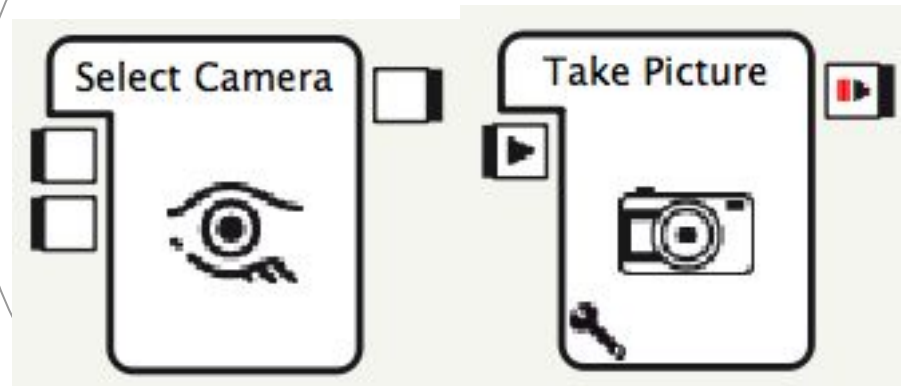
このワークショップではPepperのディレクトリ構造に詳しくなることを目指します

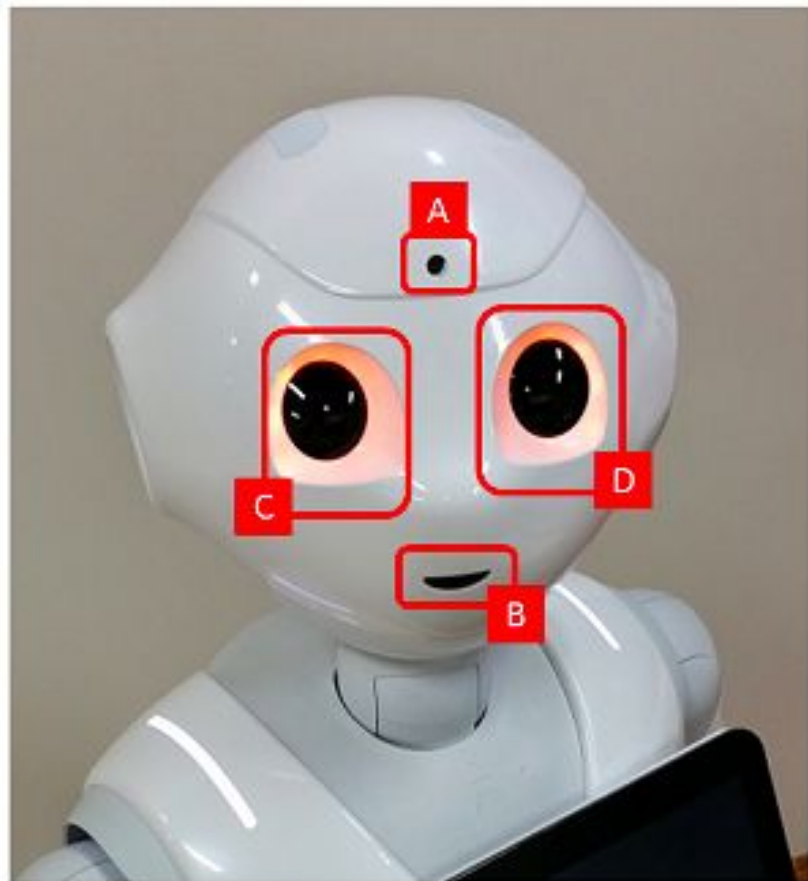


**1-a写真を撮る**



# 使うBOX

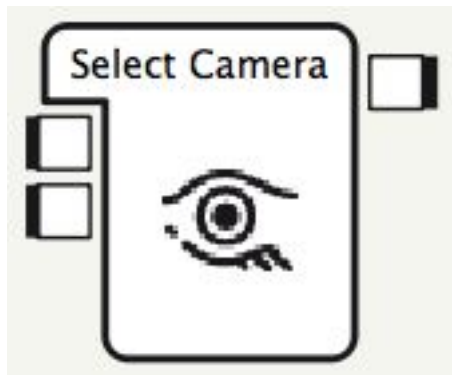




- 2Dカメラ×2 (額 [A], 口 [B])
- 3Dカメラ
  - 赤外線照射: [C]
  - 赤外線検出: [D]

# カメラを切り替えてみよう

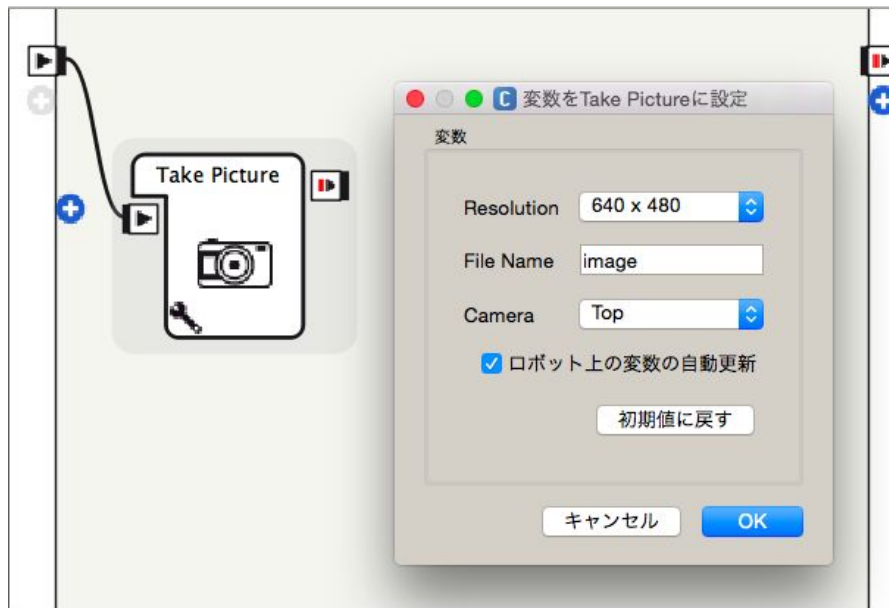
onUseTopCamera▶  
onUseBottomCamera▶  
□



- onUseTopCamera  
→ おでこのカメラを選択
- onUseBottomCamera  
→ 口元のカメラを選択

表示>  
ビデオモニター  
から確認

# 写真を撮ってみよう



## 設定ボタン

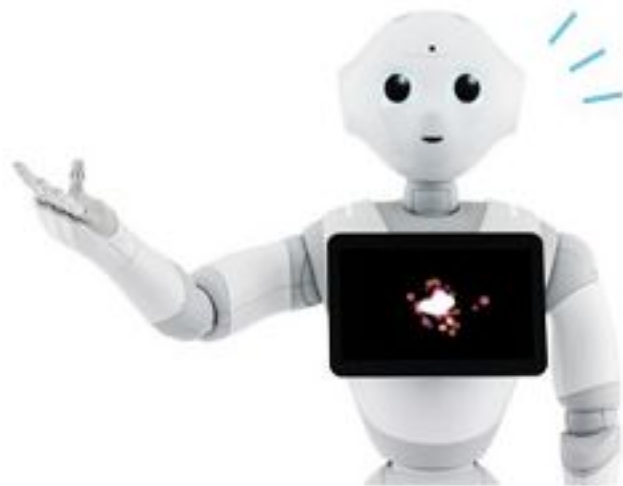


- Resolution: 解像度
- File Name: 名前
  - デフォルトでは  
"image.jpg"で保存される
- Camera: カメラの選択
  - Top: おでこのカメラ
  - Bottom: 口のカメラ

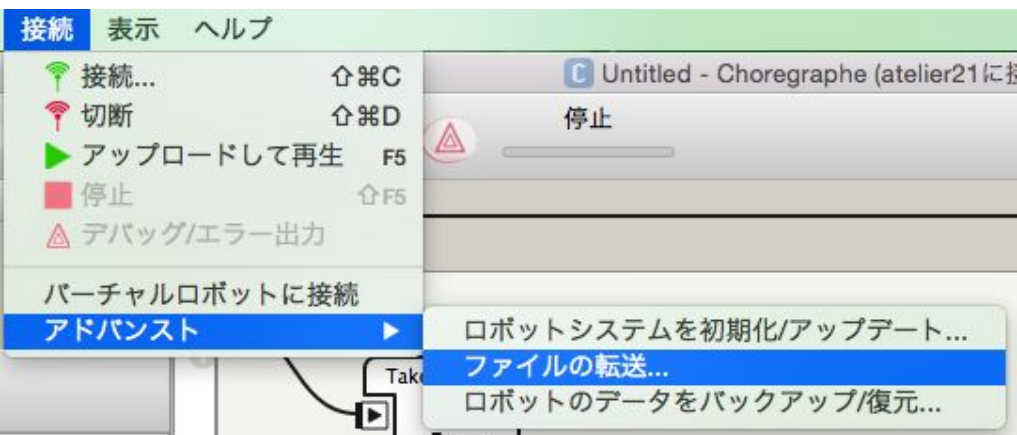
※撮った写真はPepperの中に  
保存される

/home/nao/recordings/cameras

**1-b写真をPCに  
転送**

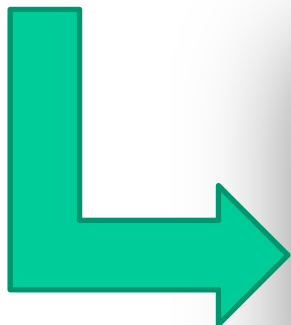
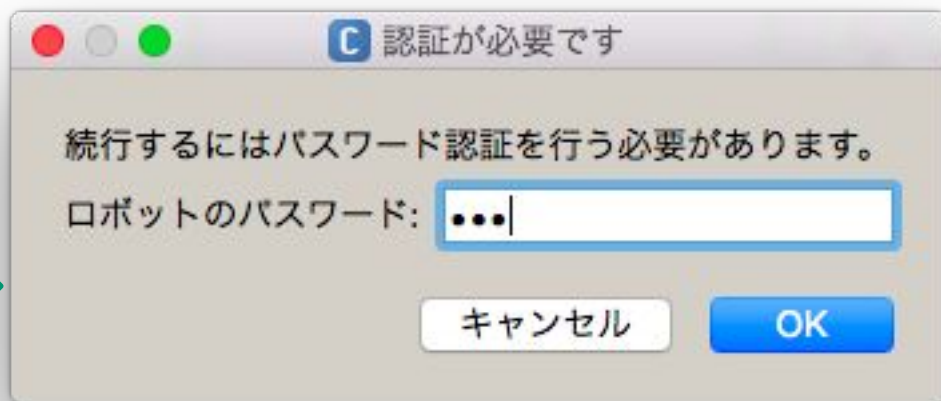


# PepperからPCへ転送



接続>アドバンスト  
>[ファイルの転送...]を選択

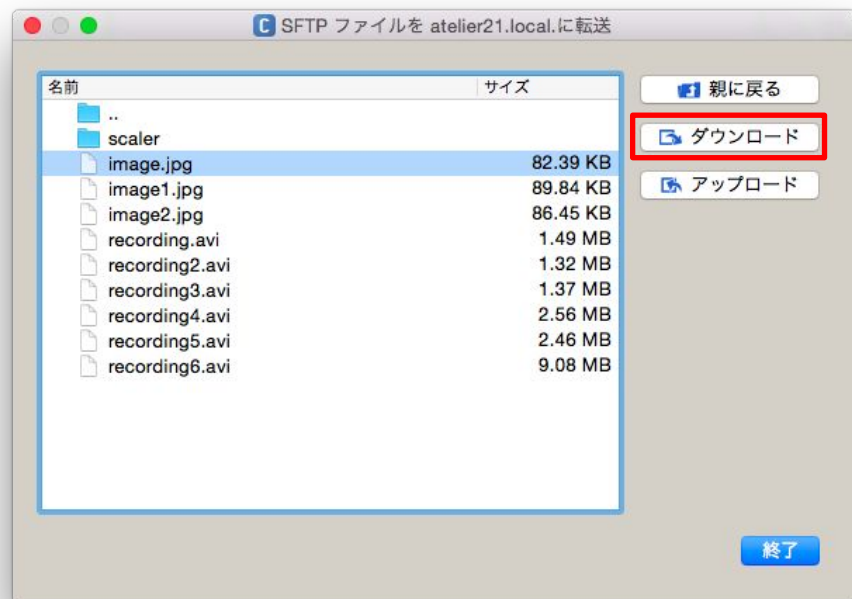
ロボットのパスワード : nao







# PepperからPCへ転送

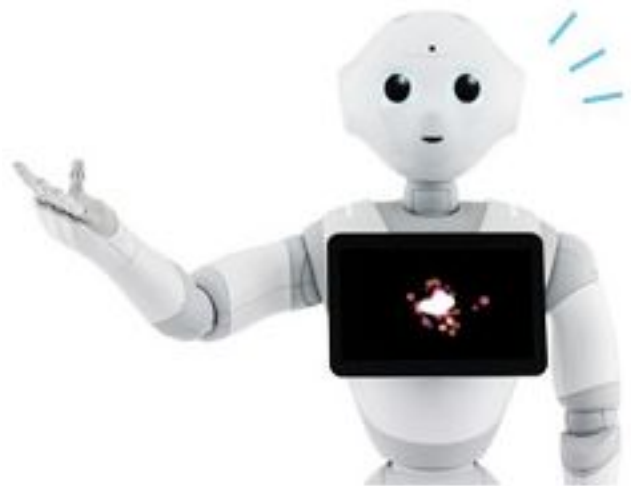
nao > recordings > cameras に指定したファイル名で保存されている



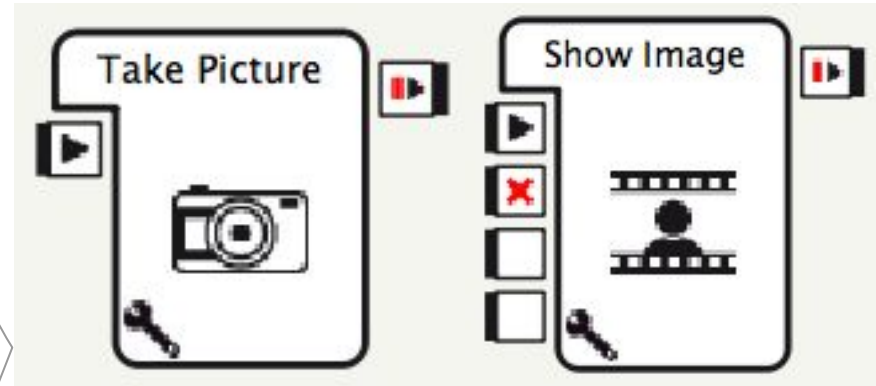
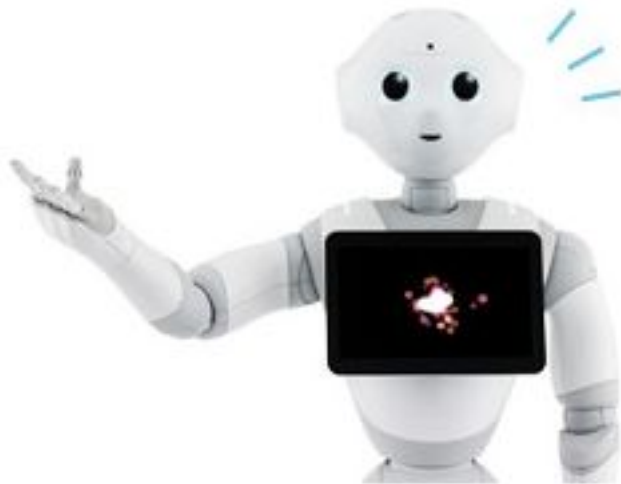
-  ダウンロード : 指定のファイルを  
PepperからPCに転送
-  アップロード : PCからPepperの  
指定のフォルダに転送

※同じファイル名で保存すると  
上書き保存される

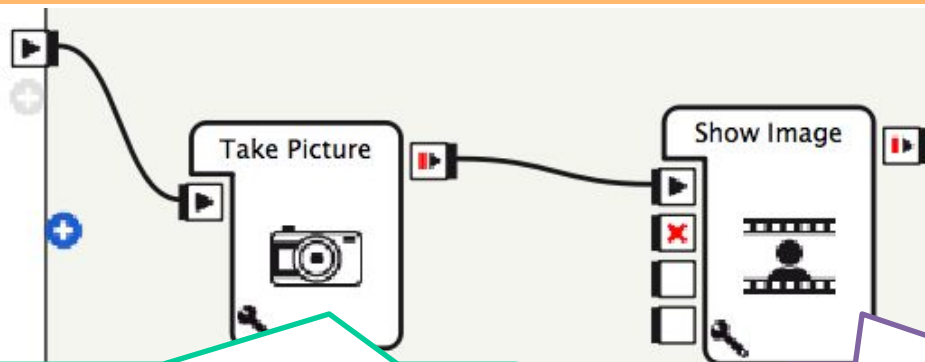
**1-c 写真をタブ  
レットに表示**



# 使うBOX



## 写真をタブレットに表示



撮った写真はPepper内  
/home/nao/recordings/cameras  
に保存されている

アプリケーションの  
htmlフォルダ内の画像を  
タブレットに表示



take picture BOXの保存先がhtmlになるように変更

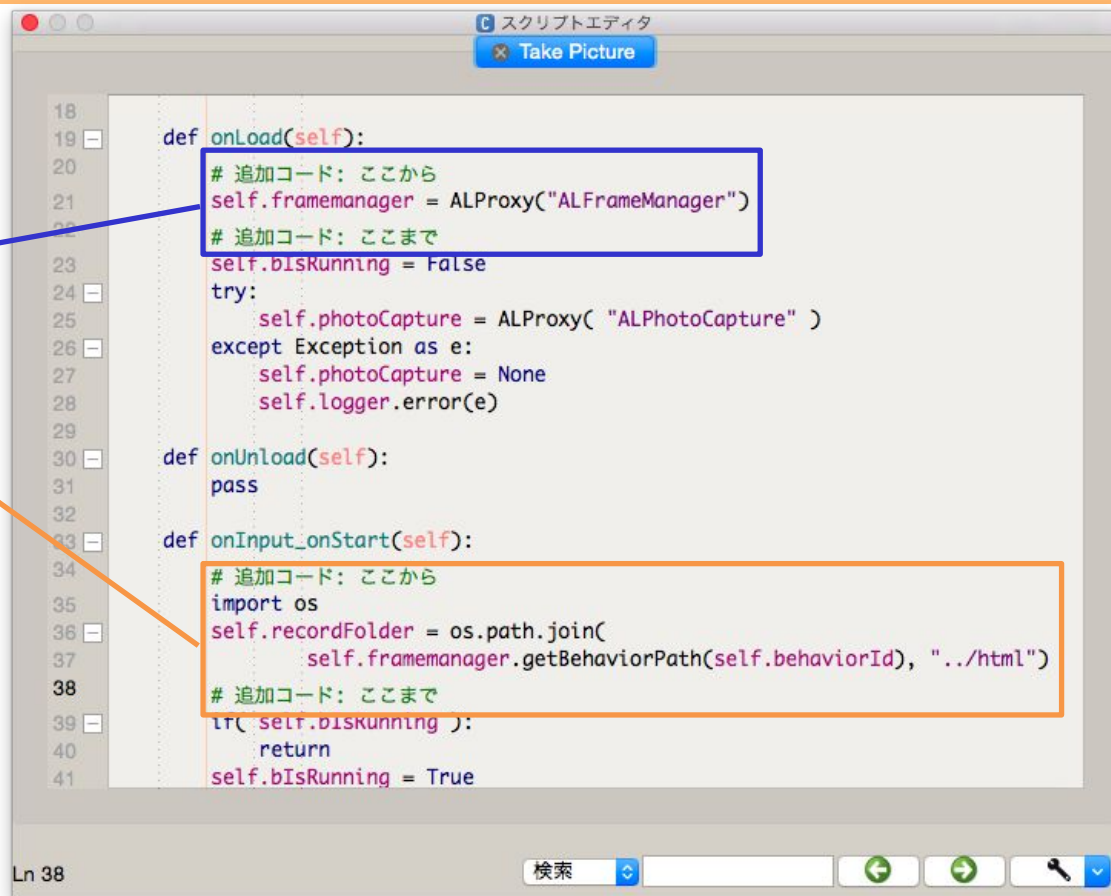
# take pictureの書き換え

ALFrameManager への  
アクセス用オブジェクトを  
生成するコードを追加

htmlディレクトリのパスを  
取得するコードを追加

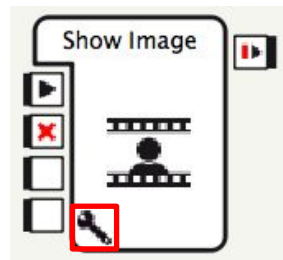
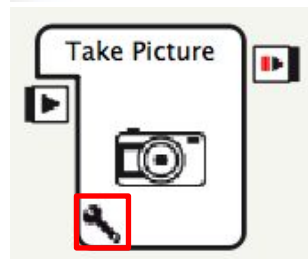
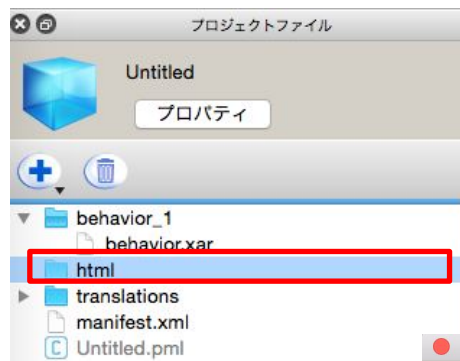
※配布資料

takepicture\_showimage.txt



```
18
19 def onLoad(self):
20     # 追加コード: ここから
21     self.frameanager = ALProxy("ALFrameManager")
22     # 追加コード: ここまで
23     self.bIsRunning = False
24     try:
25         self.photoCapture = ALProxy( "ALPhotoCapture" )
26     except Exception as e:
27         self.photoCapture = None
28         self.logger.error(e)
29
30 def onUnload(self):
31     pass
32
33 def onInput_onStart(self):
34     # 追加コード: ここから
35     import os
36     self.recordFolder = os.path.join(
37         self.frameanager.getBehaviorPath(self.behaviorId), "../html")
38     # 追加コード: ここまで
39     if self.bIsRunning ):
40         return
41     self.bIsRunning = True
```

# ファイルの設定

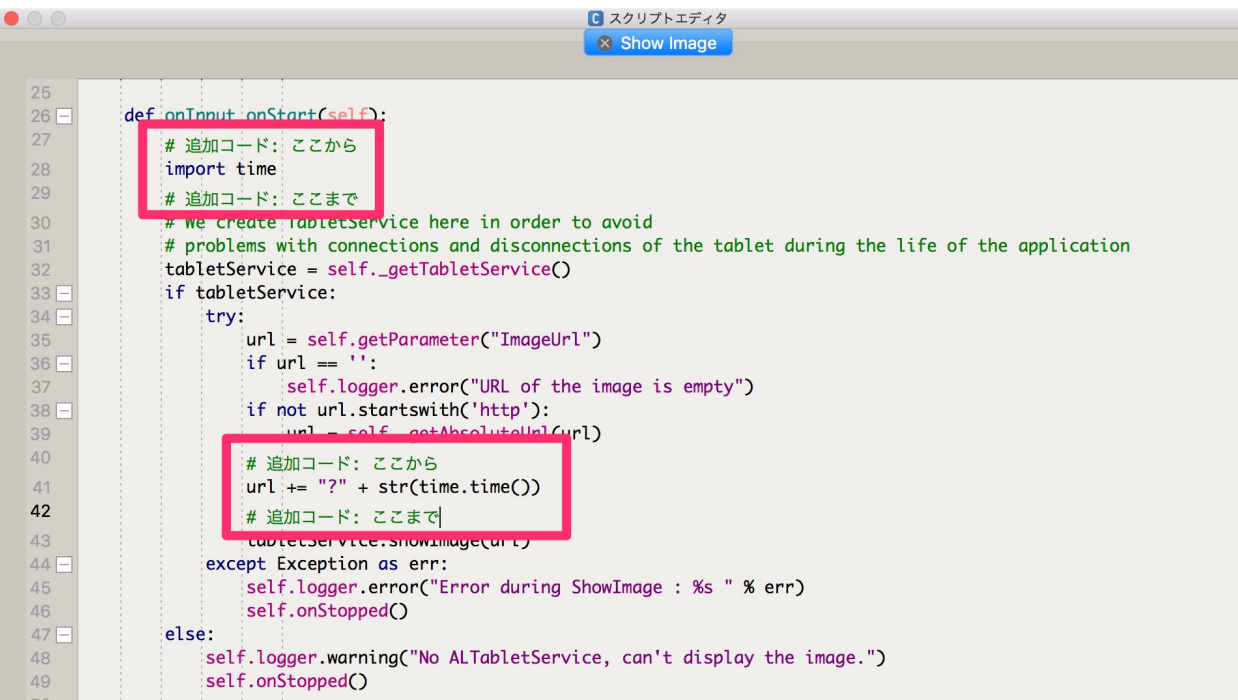


①htmlディレクトリの作成  
プロジェクトファイル  
＞ディレクトリの作成  
→ htmlディレクトリの作成  
→ダミーのimage.jpgを入れる

②Take PictureのFile Nameは  
拡張子なしでOK

③Show ImageのImageUrlは  
拡張子をつける

# タブレットのキャッシュの回避



```
25
26 def onInput_onStart(self):
27     # 追加コード: ここから
28     import time
29     # 追加コード: ここまで
30     # we create tabletService here in order to avoid
31     # problems with connections and disconnections of the tablet during the life of the application
32     tabletService = self._getTabletService()
33     if tabletService:
34         try:
35             url = self.getParameter("ImageUrl")
36             if url == '':
37                 self.logger.error("URL of the image is empty")
38             if not url.startswith('http'):
39                 url = self._getAbsolutePath(url)
40             # 追加コード: ここから
41             url += "?" + str(time.time())
42             # 追加コード: ここまで
43             tabletService.showImage(url)
44         except Exception as err:
45             self.logger.error("Error during ShowImage : %s " % err)
46             self.onStopped()
47     else:
48         self.logger.warning("No ALTabletService, can't display the image.")
49         self.onStopped()
50
```

タブレットの内容が更新されないという現象が発生します。

→Webブラウザが画像をキャッシュすることが原因

→実行時の時間をURLのクエリに含めてあげることで解決

**SSHでPepperに  
接続**





# SSHとは？

- Secure Shell(セキュア シェル)の略
- 暗号や認証の技術を利用して、安全にリモートコンピュータと 通信するためのプロトコル
  - パスワードやデータを暗号化して通信
  - クライアントがサーバーに接続する時に、接続先が意図しない サーバーに誘導されていないか厳密にチェックする

SSHを利用して、Choregrapheを経由せずに  
PepperとPCを接続しよう！

- sshコマンド
  - リモートホストに接続してコマンドを実行
  - `$ ssh [user@]hostname [command]`
- scpコマンド
  - SSHを使用してリモートホストとの間でファイルの転送を行う
  - `$ scp file1 file2 [user@]hostname[:dir1]`
  - `$ scp [user@]hostname:file1 [user@]hostname:file2 dir1`

## 実際に接続してみよう

- ターミナルかコマンドプロンプトからsshコマンドを実行
- `$ ssh [user@]hostname [command]`
  - ユーザ名 : `nao@`
  - ホスト名 : `[Pepperの名前].local` もしくは IPアドレス
- 例 : `brassy254`に接続

```
$ ssh nao@brassy254.local
```

atelier43に接続

```
$ ssh nao@atelier43.local.
```

```
$ ssh nao@192.168.100.**
```

## 中身を見てみよう

- 中身を見るのに使うコマンド

- `$ pwd`: 今いる地点の表示

```
P4B03 [0] ~ $ pwd  
/home/nao
```

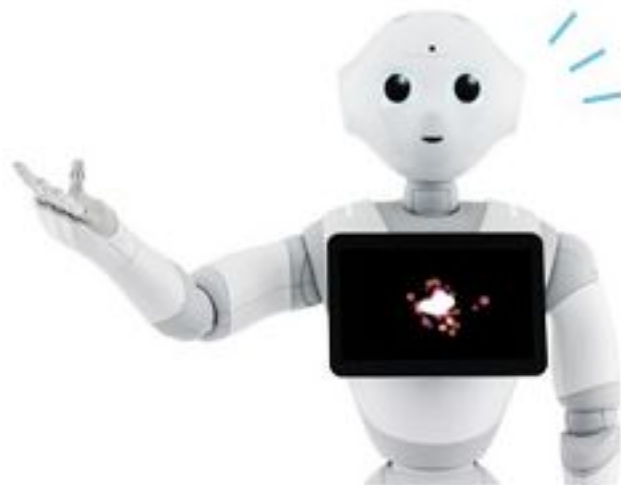
- `$ ls`: 今いる地点にあるファイルを一覧表示

```
P4B03 [0] ~ $ ls  
MyCustomPepperService.py  diagnosis  naoqi  recordings
```

- `$ cd [ディレクトリ名]: [ディレクトリ名]に移動`

```
P4B03 [0] ~ $ cd recordings/  
P4B03 [0] ~/recordings $ pwd  
/home/nao/recordings
```

**SSHで動かして  
みよう**



- Pepperの中にインストール済みのユーティリティプログラム
- qicliでできること

qicli info

サービスの情報を取得する

qicli call

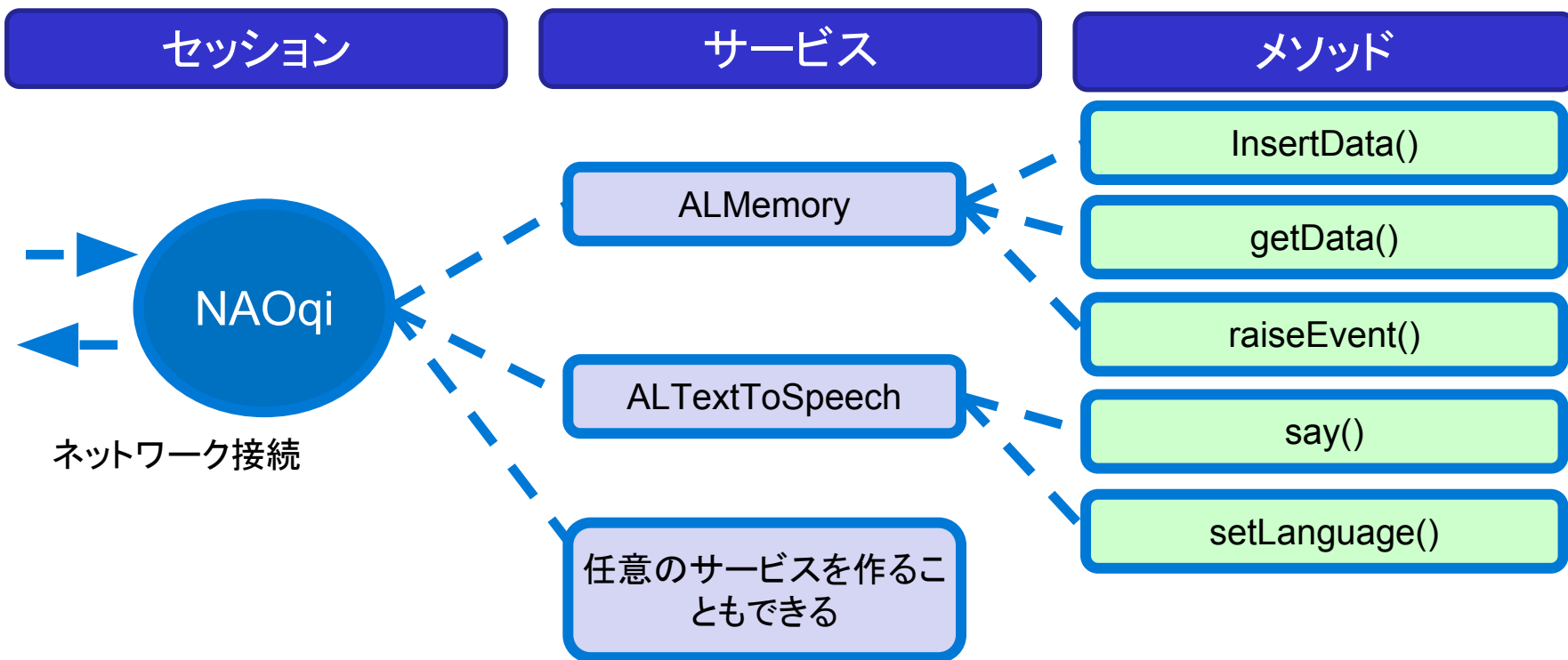
APIを呼び出す

qicli watch

イベントを監視する

qicli log-view

ログを監視する



qicli info でサービスの情報を取得

- サービス一覧の表示

```
$ qicli info
```

- サービスの API 一覧の表示

```
$ qicli info <サービス名>
```

qicli watch でシグナルやイベントのリアルタイム監視ができる

- シグナルの監視

```
$ qicli watch <サービス名.シグナル名>
```

- ALMemoryイベントの監視

```
$ qicli watch --almemory <ALMemoryキー名>
```



- qicli call で API を呼び出すことができる

```
$ qicli call <サービス名>.<API名> <引数>
```

- 例: Pepperに「こんにちは」と言ってほしい
  - APIモジュール: ALTextToSpeech
  - メソッド: say
  - パラメータ: 話させたい文字列("hello")

```
$ qicli call ALTextToSpeech.say hello
```

- **音量設定の表示**

```
$ qicli call ALAudioDevice.getOutputVolume
```

- **バッテリーの表示**

```
$ qicli call ALBattery.getBatteryCharge
```

- **Wi-fi 通信状態の表示**

```
$ qicli call ALTabletService.getWifiStatus
```

- **実行中のアプリの表示**

```
$ qicli call ALAutonomousLife.focusedActivity
```

- **音量設定の変更**

```
$ qicli call ALAudioDevice.setOutputVolume 40
```

- **移動**

```
$ qicli call ALMotion.moveToward 0 1 0
```

- **対象アプリの起動**

```
$ qicli call ALAutonomousLife.switchFocus <アプリID>/.
```

**SSHからアプリ実  
行**




- アプリケーションの実行
  - APIモジュール: ALAutonomousLife
  - メソッド: switchFocus
  - パラメータ: [アプリケーションID/Behaviorフォルダ]
  - Behaviorフォルダ: Behavior.xarが入っているフォルダ
- 例: 設定アプリの実行

```
$ qicli call ALAutonomousLife.switchFocus boot-config/.
```

# ビヘイビアパスの確認

- アプリケーションID一覧
  - .local/share/PackageManager/apps/

```
Brassy254 [0] ~ $ ls .local/share/PackageManager/apps/  
_schfes_end  
aaa_360_bz10002_nutcracker_v  
aaunin-safe  
aic  
aiclient_subsystem  
altactilegesture-9dd1d7  
animationMode  
animations  
applaucher  
asleep-update  
autonomous-recharge  
be-quiet-b2c  
boot-config  
dance_template_framework
```



設定アプリのID  
boot-config

## ビヘイビアパスの確認

- 設定アプリ(boot-config)のBehaviorフォルダ
  - behavior.xarが入っているフォルダ

```
Brassy254 [0] ~ $ ls .local/share/PackageManager/apps/boot-config/  
animations  behavior.xar  bin  html  icon.png  logs  manifest.xml  
services  sounds
```

boot-config自身に入っている  
表し方: .(カレントディレクトリ)

- 設定アプリのビヘイビアパス
  - boot-config/.

**Pepperのログ**





qicli log-view でログのリアルタイム監視ができる

- ログのリアルタイム監視

```
$ qicli log-view  
$ tail -f /var/log/naoqi/head-naoqi.log
```

- ログレベルを指定したログ表示

```
$ qicli log-view -l <引数>
```

- ログ情報のフィルタ設定

```
$ qicli log-view -f <引数>
```

おまけ





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

# 開発のための情報



 Pepper フォロー中

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  
AWS IoT で Pepper と RaspberryPi 間を MQTT でやり取りしてみる


6

 Python  RaspberryPi  Pepper  awsIoT



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





18

 Pepper  Choregraphe



yuka\_nm が2016/09/23に投稿  
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






26

 やってみた  Pepper  Watson  SpeechToText



JohnTomato が2015/12/15に投稿  
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

52

 AdventCalendar  Bluemix  NAOqi  Pepper  Watson



kakkey が2016/08/23に投稿  
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



1

 NAOqi  Pepper



kakkey が2016/08/23に投稿  
Pepperのセンサーを長押ししてイベント発火させるボックスを作成してみた

5

 NAOqi  Pepper



a-r-i が2018/05/07に投稿  
ペッパーくんをアイドルオタク化してみた

6

Qiita

「Qiita pepper」で検索

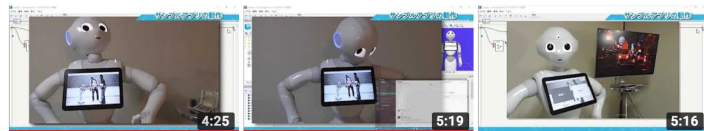
・プログラミングに関する知識を  
記録・共有するためのサービス

・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

≡ 並べ替え



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



第14回 Pepperからスマホに通知を送ってみよう！(スマ...

視聴回数 63 回・7 か月前

第13回 Pepperで照明を制御！(Hue編)

視聴回数 17 回・7 か月前

第12回 Pepperでエアコンを制御してみよう！(iRemoco...

視聴回数 52 回・7 か月前

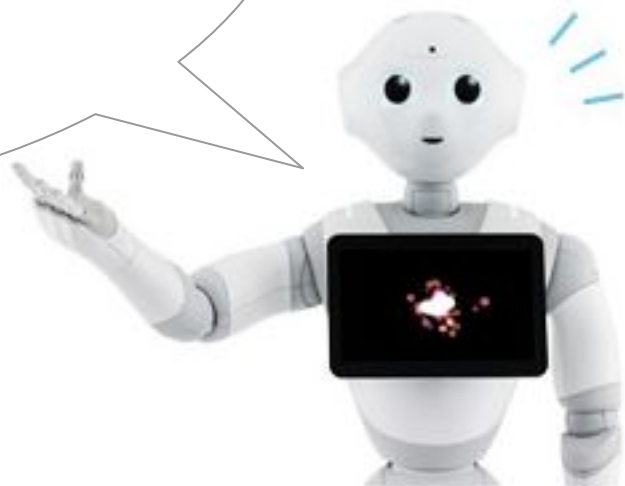
## Pepper Developer Network

「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

**おつかれさまでした！  
これにてWSは終わりになります。**



アンケートへのご協力お願いします

<https://bit.ly/pepperatelier>

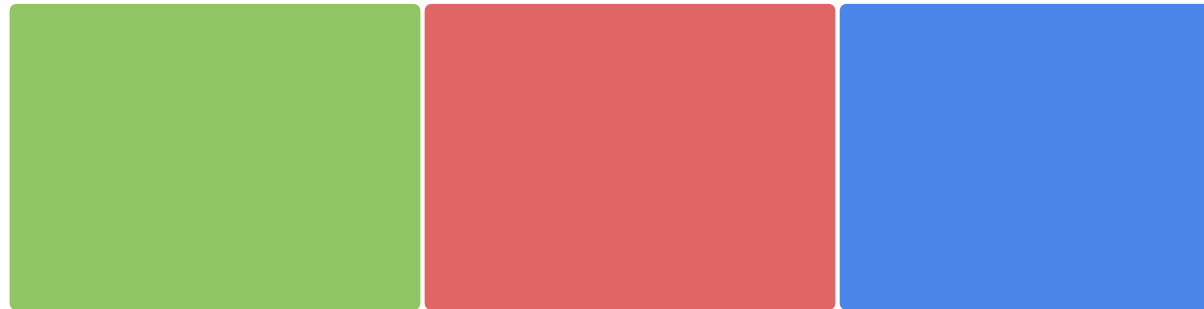




# マーカーの色の参考にしてください(改変OK)

↓※開発中は停止しておく

プログラムの再生↓



接続↑

↑切断

てすとtext

てすとtext