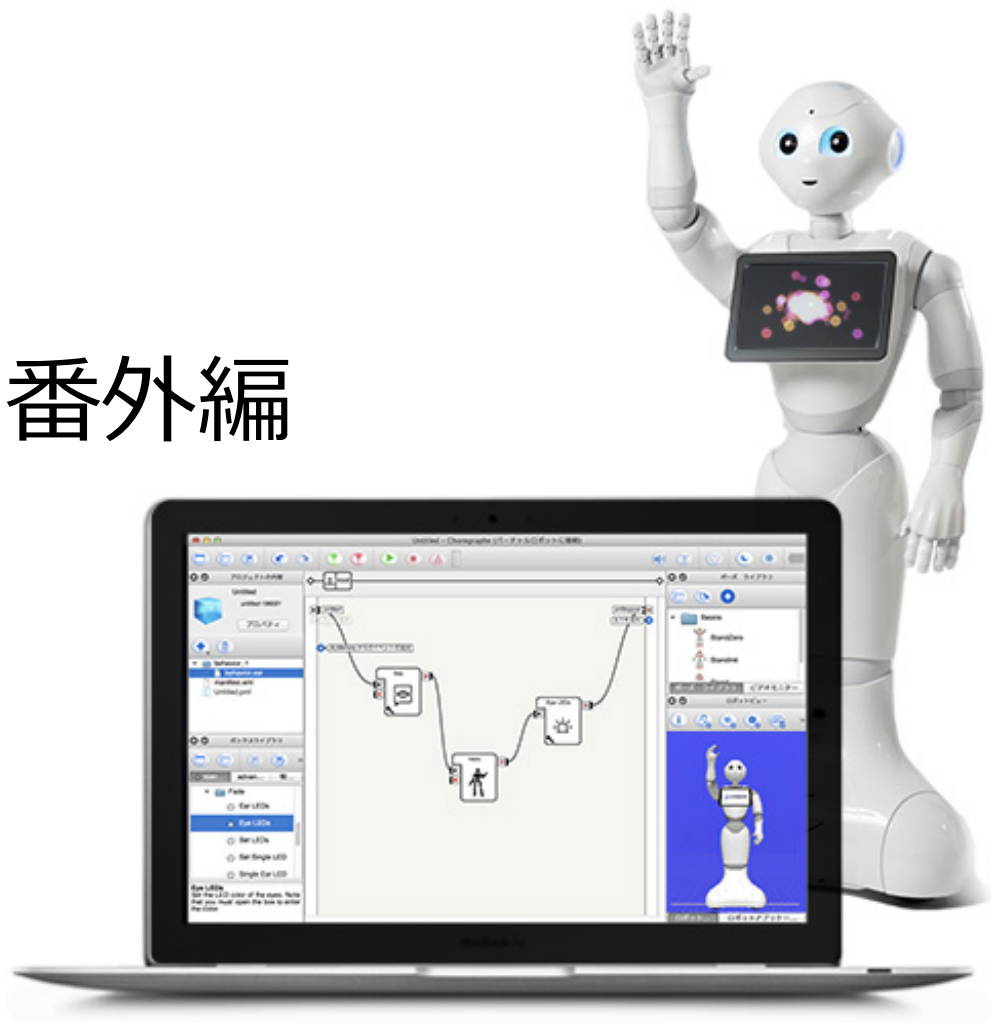


Atelier Akihabara

ワークショップ 番外編 API 入門

アトリエ秋葉原 Presents.



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

アトリエについて

実体験とコミュニティで開発を促進する

アトリエ



✓ アトリエサテライト
全国各地で発足中

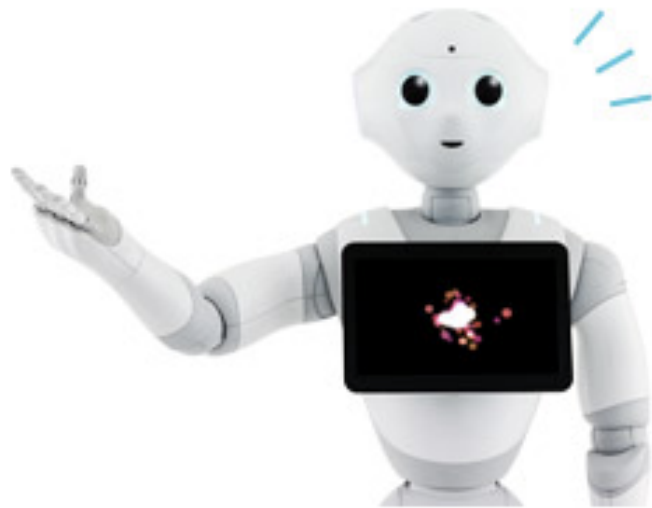
コミュニティ



✓ デベロッパーによる勉強会

相互
促進

自己紹介
してみませんか？



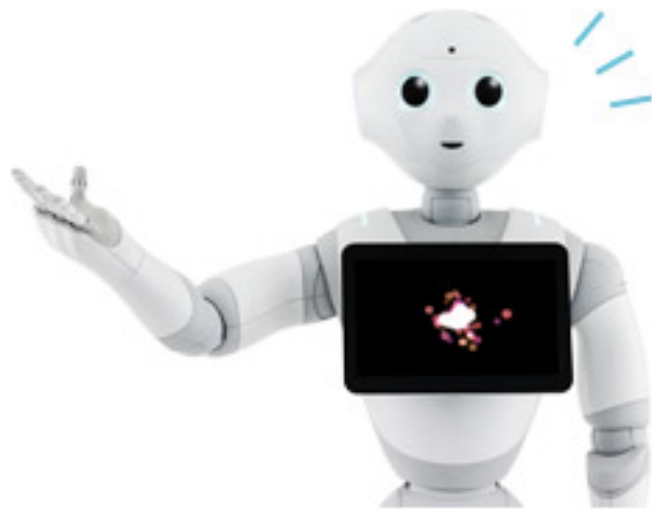
- ・ **お名前**
- ・ **ご所属**
- ・ **本日の意気込み**

例：

本日の案内を勤めさせていただきます、
松尾 映里 (Matsuo Eri) と申します。

本WSの内容

1. APIとは？
2. PepperでWebサービスを利用
3. お天気API で天気を取得
4. twitter API でツイートを投稿
5. おまけ



API とは？

API (Application Programming Interface)

- 中身(仕組み)は公開されていないが、**機能だけ利用できるサービス**
 - 必要な情報を渡せば、難しいことをせずに便利なサービスを利用できる！
 - 「**渡す情報**」「**渡す情報の形式**」「**得られる情報の形式**」はAPIごとに異なる
 - 多くのAPIは利用のための Keyとパスワードの用意が必要
- 例1：天気予報のAPIに、知りたい「場所」「時間」を渡せば、
（仕組みはわからないが）天気を教えてくれる
- 例2：twitterのAPIに、投稿したい「アカウント」「ツイート内容」を渡せば、
（仕組みはわからないが）twitterに投稿してくれる

例1 : Livedoor お天気Webサービス

- http://weather.livedoor.com/weather_hacks/webservice
 - 公式ページ



お天気Webサービス（Livedoor Weather Web Service / LWWS）は、現在全国142カ所の今日・明日・あさっての天気予報・予想気温と都道府県の天気概況情報を提供しています。

● このAPIの仕様

- 渡す情報 : 地域
- 渡す情報の形式 : 指定URLの末尾に地域IDを加える
- 得られる情報の形式 : JSONデータ（細かい形式は上記サイト参照）
- API keyは不要（URLに接続すれば誰でも情報取得可能）

※ これらはAPI提供側が必ず公開している必須情報！

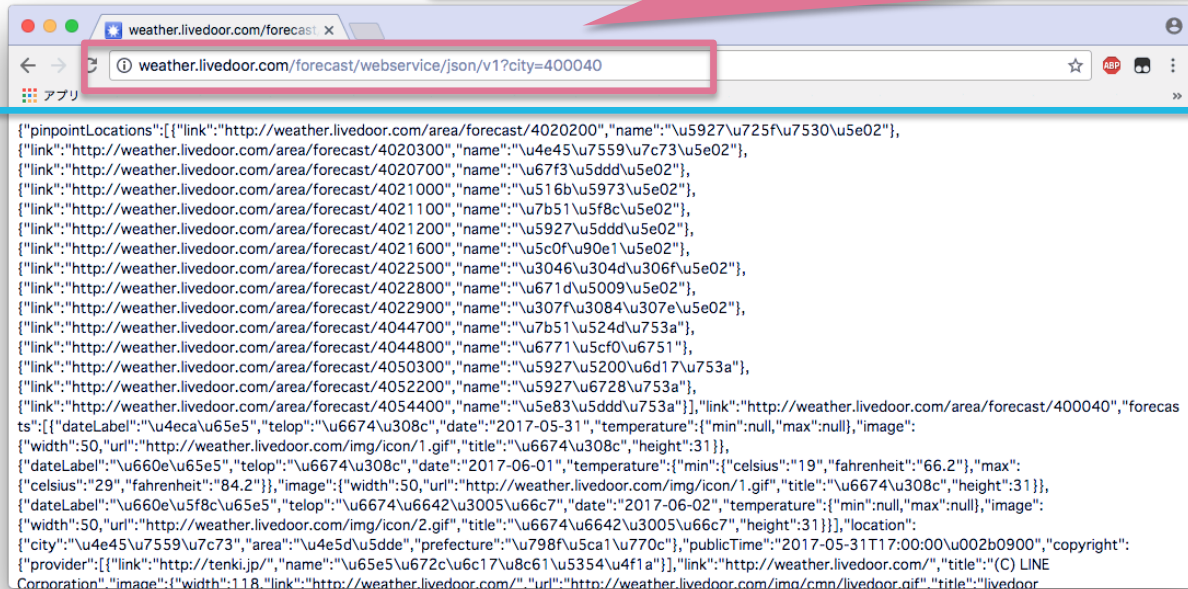
APIの例

例1 : Livedoor お天気Webサービス

- http://weather.livedoor.com/weather_hacks/webservice

<http://weather.livedoor.com/forecast/webservice/json/v1?city=400040>

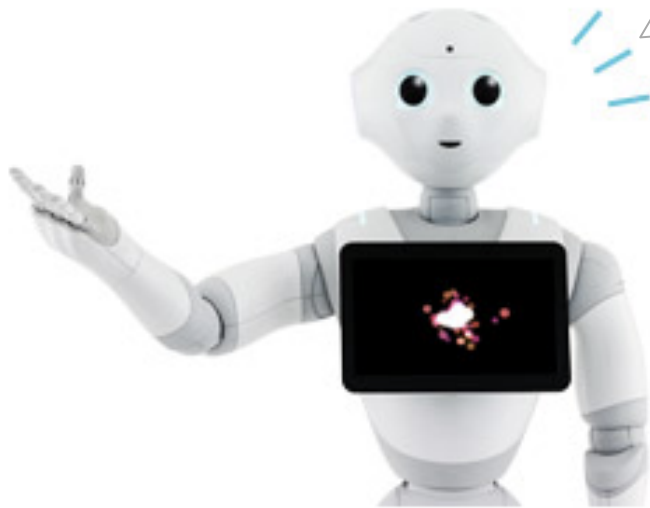
↑
久留米市のID



```
{
  "pinpointLocations": [
    {
      "link": "http://weather.livedoor.com/area/forecast/4020200",
      "name": "\u5927\u725f\u7530\u5e02",
      "link": "http://weather.livedoor.com/area/forecast/4020300",
      "name": "\u4e45\u7559\u7c73\u5e02",
      "link": "http://weather.livedoor.com/area/forecast/4020700",
      "name": "\u67f3\u5ddd\u5e02",
      "link": "http://weather.livedoor.com/area/forecast/4021000",
      "name": "\u516b\u5973\u5e02",
      "link": "http://weather.livedoor.com/area/forecast/4021100",
      "name": "\u7b51\u5f8c\u5e02",
      "link": "http://weather.livedoor.com/area/forecast/4021200",
      "name": "\u5927\u5ddd\u5e02",
      "link": "http://weather.livedoor.com/area/forecast/4021600",
      "name": "\u5c0f\u90e1\u5e02",
      "link": "http://weather.livedoor.com/area/forecast/4022500",
      "name": "\u3046\u304d\u306f\u5e02",
      "link": "http://weather.livedoor.com/area/forecast/4022800",
      "name": "\u671d\u5009\u5e02",
      "link": "http://weather.livedoor.com/area/forecast/4022900",
      "name": "\u307f\u3084\u307e\u5e02",
      "link": "http://weather.livedoor.com/area/forecast/4044700",
      "name": "\u7b51\u524d\u753a",
      "link": "http://weather.livedoor.com/area/forecast/4044800",
      "name": "\u671\u5cf0\u6751",
      "link": "http://weather.livedoor.com/area/forecast/4050300",
      "name": "\u5927\u5200\u617\u753a",
      "link": "http://weather.livedoor.com/area/forecast/4052200",
      "name": "\u5927\u6728\u753a",
      "link": "http://weather.livedoor.com/area/forecast/4054400",
      "name": "\u5e83\u5ddd\u753a",
      "link": "http://weather.livedoor.com/area/forecast/400040",
      "forecast": {
        "dateLabel": "\u4eca\u65e5",
        "telop": "\u6674\u308c",
        "date": "2017-05-31",
        "temperature": {
          "min": null,
          "max": null,
          "image": {
            "width": 50,
            "url": "http://weather.livedoor.com/img/icon/1.gif",
            "title": "\u6674\u308c",
            "height": 31
          },
          "dateLabel": "\u660e\u65e5",
          "telop": "\u6674\u308c",
          "date": "2017-06-01",
          "temperature": {
            "min": {
              "celsius": "19",
              "fahrenheit": "66.2",
              "max": {
                "celsius": "29",
                "fahrenheit": "84.2"
              },
              "image": {
                "width": 50,
                "url": "http://weather.livedoor.com/img/icon/1.gif",
                "title": "\u6674\u308c",
                "height": 31
              }
            },
            "dateLabel": "\u660e\u5f8c\u65e5",
            "telop": "\u6674\u3005\u66c7",
            "date": "2017-06-02",
            "temperature": {
              "min": null,
              "max": null,
              "image": {
                "width": 50,
                "url": "http://weather.livedoor.com/img/icon/2.gif",
                "title": "\u6674\u3005\u66c7",
                "height": 31
              },
              "location": {
                "city": "\u4e45\u7559\u7c73",
                "area": "\u4e5d\u5ddd",
                "prefecture": "\u798f\u5c11\u770c",
                "publicTime": "2017-05-31T17:00:00\u002b0900",
                "copyright": {
                  "provider": [
                    {
                      "link": "http://tenki.jp/",
                      "name": "\u65e5\u672c\u6c1\u7861\u5354\u4f1a",
                      "link": "http://weather.livedoor.com/",
                      "title": "(C) LINE Corporation",
                      "image": {
                        "width": 118,
                        "link": "http://weather.livedoor.com/",
                        "url": "http://weather.livedoor.com/img/cmp/livedoor.gif",
                        "title": "livedoor"
                      }
                    }
                  ]
                }
              }
            }
          }
        }
      }
    }
  ]
}
```

天気情報を表すJSONデータ

Pepperで
Webサービスを利用



PepperからWeb接続

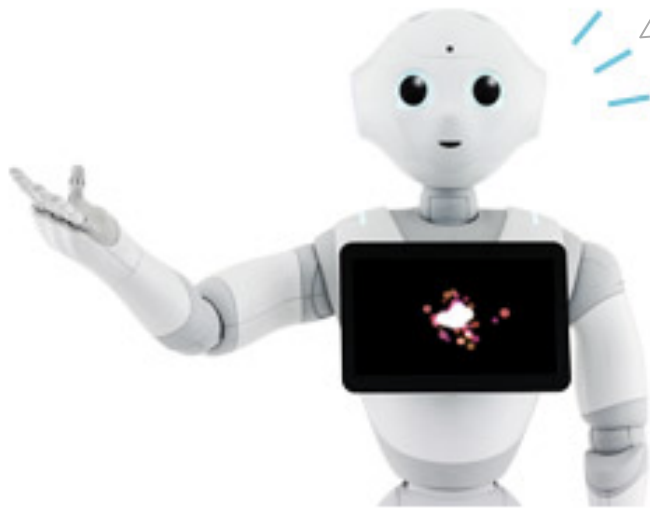
- ChoregrapheからWebにアクセス
 - 今回は**Boxスクリプト(=Python)**内から接続したい
- 例：「**requests**」ライブラリを使用
 - Python (=Box) で**HTTP通信**を行うライブラリ
 - 他、「urllib」「httplib2」など様々



```
11 pass
12
13 def onInput_onStart(self):
14 |
15 |     import requests
16 |     url = "http://weather.livedoor.com/forecast/webservice/json/v1?city=400040"
17 |     r = requests.get(url)
18
```



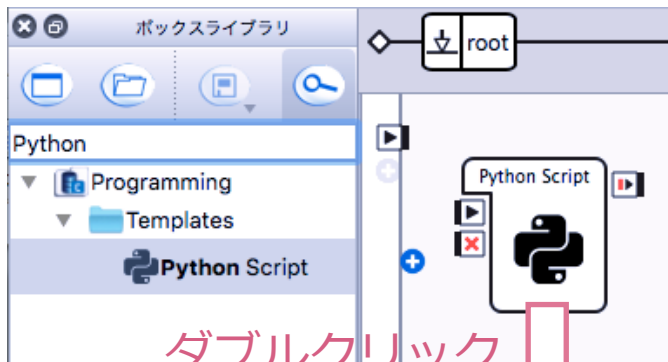
お天気 API で
天気を取得



お天気 API で天気を取得

- Livedoor お天気Webサービスを使って、
今日の天気を Pepper が喋るアプリをつくろう！
- 手順
 - お天気APIから情報を取得し、テキスト(文字列)にするBoxを作成
 - requestsライブラリで**Webにアクセス**する
 - 天気情報を処理し、喋らせたい**文字列に変換**する
 - 受け取ったJSONデータはそのままでは使えない！
 - **文字列を出力**させる
 - 「Say Text」Boxで**発話**

お天気 API で天気を取得



① 「Python Script」 Box をパネルに出す

② Python Script をダブルクリックし、
Boxの中身を変更

15 | pass を削除し、そこに以下をコピー (tenki_script.txt)

```
import requests
```

```
url = "http://weather.livedoor.com/forecast/webservice/json/v1?city=130010"
```

```
r = requests.get(url)
```

```
data = r.json()
```

↑ 東京のID

```
ken = data['location']['prefecture'].encode('utf8') # Tokyo-to
```

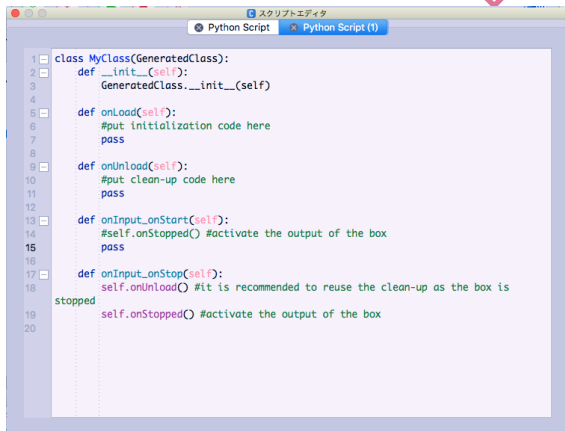
```
shi = data['location']['city'].encode('utf8') # Tokyo
```

```
date = data['forecasts'][0]['dateLabel'].encode('utf8') # today
```

```
tenki = data['forecasts'][0]['telop'].encode('utf8') # weather
```

```
tenki_text = ken + shi + 'の' + date + 'の天気は' + tenki + 'です。'
```

```
self.text(tenki_text)
```



お天気 API で天気を取得 Boxの書き換え

15行目以降を変更

15

```
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
        stopped
        self.onStopped() #activate the output of the box
```

tabキーやスペースキーで
字下げ を必ず揃える！

```
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
        import requests
        url = "http://weather.livedoor.com/forecast/webservice/json/v1?city=130010"
        r = requests.get(url)
        data = r.json()

        ken = data['location']['prefecture'].encode('utf8') # Tokyo-to
        shi = data['location']['city'].encode('utf8') # Tokyo
        date = data['forecasts'][0]['dateLabel'].encode('utf8') # today
        tenki = data['forecasts'][0]['telop'].encode('utf8') # weather
        tenki_text = ken + shi+'の'+date+'の天気は'+tenki+'です。'
        self.text(tenki_text)

    def onInput_onStop(self):
```

お天気 API で天気を取得 Boxの出力作成

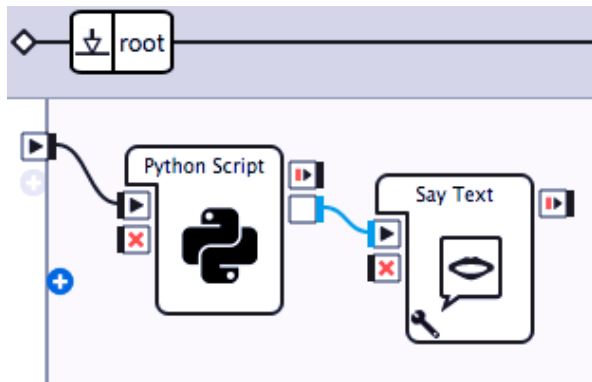
- お天気APIから取得した情報をPepperに喋らせたい！
 - 「Python Script」 Box から「Say Text」 Box に読んでほしいテキストを渡す
- 新しく「文字列を渡す出力」をつくる！

③ボックスの出力周辺を右クリック、
[出力を追加]を選択

④名前:「text」
型:「文字列」
に変更し、OKを選択



お天気 API で天気を取得

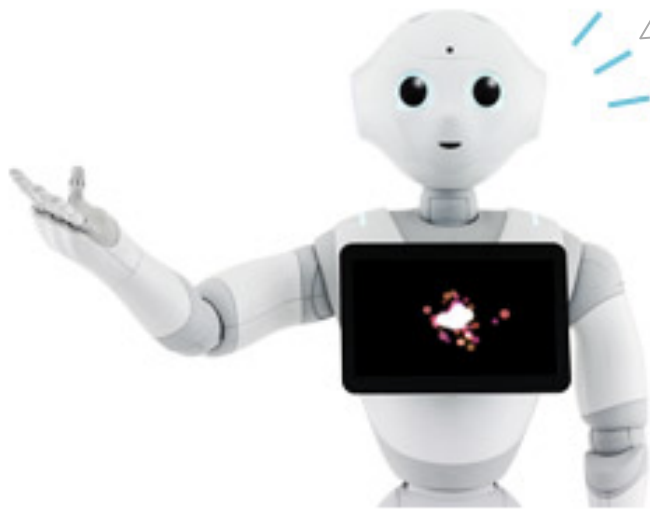


⑤左のようにボックスを配置して完成！

おまけ：Livedoor お天気Webサービスの仕様を参考にスクリプトを変更してみよう

- http://weather.livedoor.com/weather_hacks/webservice
- **例1：場所を変更する**
 - url末尾「city=130010」を別の場所のIDにする
- **例2：明日の天気に変更する**
 - 「data = …」 「tenki = …」 内の [0] を [1] にする
- **例3：天気概要を喋らせる**
 - 「text = ken + shi+『の'+date+『の天気は'+tenki+『です。』」 を
「text = data['description']['text'].encode('utf8')」 に変更

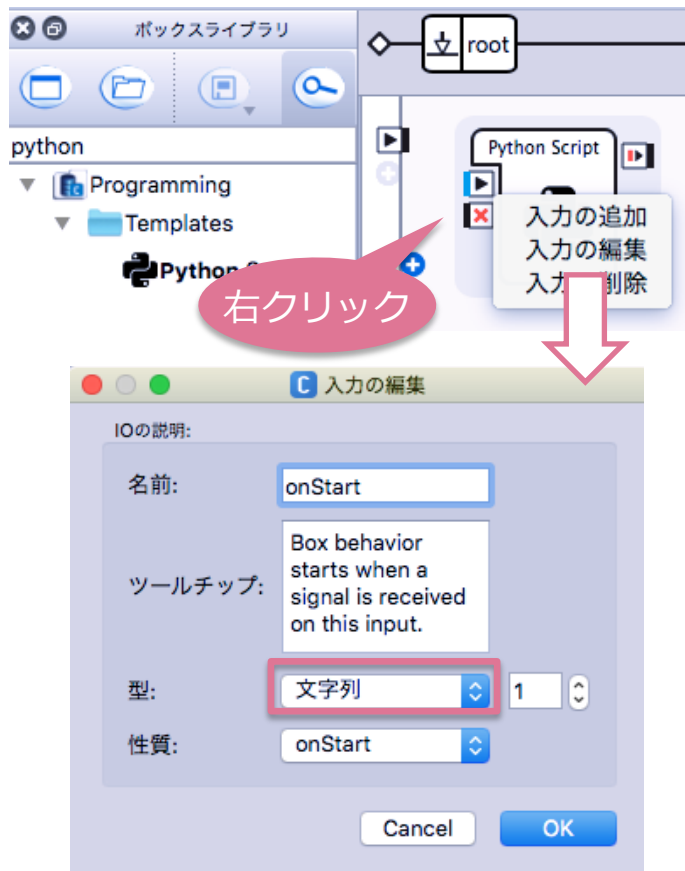
twitter API で
ツイートを投稿



Twitter API でツイート

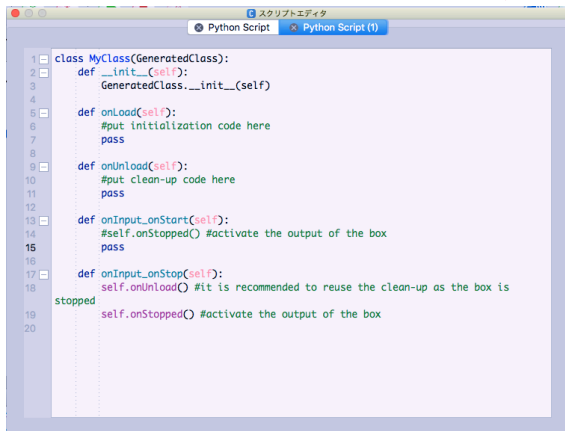
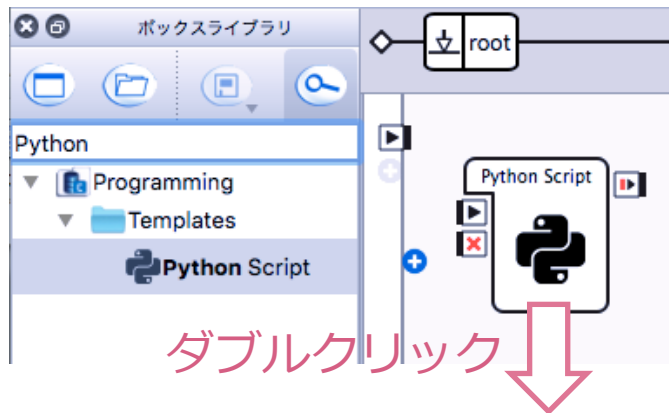
- Twitter APIを使って、
今日の天気を Pepper が **Tweet するアプリ**をつくらう！
- 手順
 - お天気APIから文字列にするBoxを再利用
 - Tweetを投稿するBoxを作成
 - 文字列を受け取る
 - 「requests_oauthlib」ライブラリでWebにアクセスする
 - twitter API は HTTPではなくOAuth認証 を使用するため
 - **外部ライブラリ(モジュール)を追加**する必要あり！
 - Tweetするアカウントの情報を入れる
 - 今回投稿するアカウント：<https://twitter.com/pepperAPItest>

文字列を入力とする



- ① 新たな「Python Script」Box をパネルに出す
- ② Boxの入力周辺を右クリック、**[入力の編集]**を選択
- ③ 型:「**文字列**」に変更し、OKを選択

文字列を入力とする



④Python Script をダブルクリックし、
Boxの中身を変更

13| def onInput_onStart(self):



13| def onInput_onStart(self, p):

↑ Boxに入力した文字列

※②～③の操作で自動的に変更されている場合もあります。
その場合は変更不要です。

外部モジュールのインポート

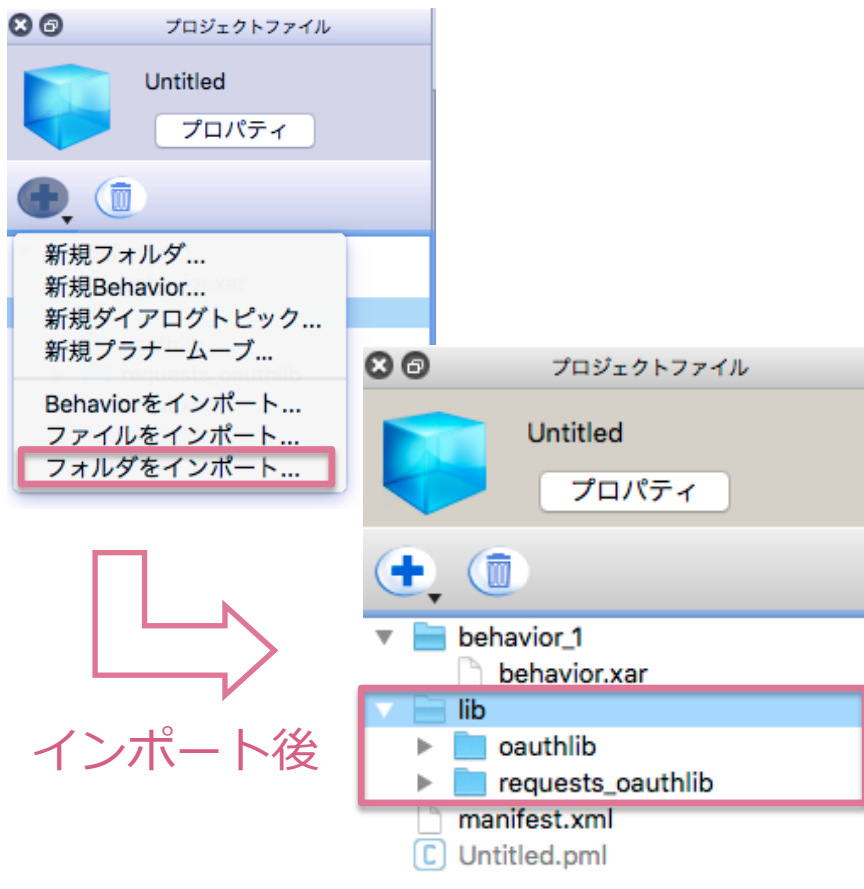
- 「requests_oauthlib」 ライブラリをインポートする必要がある
 - Twitter API のプロトコルは、HTTP ではなく OAuth
 - 「requests」ではなく「requests_oauthlib」を使いたい
- NAOqi には「requests」はインストールされているが、
「requests_oauthlib」はインストールされていない

→ **プロジェクトに外部ライブラリをインポートする！**

今回は以下の2つのライブラリをインポート

- requests_oauthlib
- oauthlib

外部モジュールのインポート



⑤(+)ボタンをクリック、
[フォルダをインポート]を選択

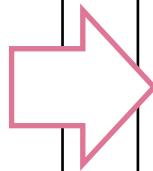
⑥配布資料内のフォルダ「**lib**」を選択

※libの内容は以下2フォルダ = ライブラリ
lib/oauthlib
lib/requests_oauthlib

外部モジュールのインポート

⑦Python Script をダブルクリックし、Boxの中身を変更 (lib/の下にあるライブラリを読み込むようにする)

```
5| def onLoad(self):  
    #put initialization code here  
    pass  
  
9| def onUnload(self):  
    #put clean-up code here  
    pass  
  
13| def onInput_onStart(self, p):  
    #self.onStopped() #activate the output of the box  
    pass
```



```
def onLoad(self):  
    self.framemanager = ALProxy("ALFrameManager")  
    self.folderName = None  
  
def onUnload(self):  
    import sys  
    if self.folderName and self.folderName in sys.path:  
        sys.path.remove(self.folderName)  
    self.folderName = None  
  
def onInput_onStart(self, p):  
    import sys, os  
    self.folderName = os.path.join(self.framemanager.get  
BehaviorPath(self.behaviorId), "../lib")  
    if self.folderName not in sys.path:  
        sys.path.append(self.folderName)
```

外部モジュールのインポート

⑦Python Script をダブルクリックし、Boxの中身を変更 (lib/の下にあるライブラリを読み込むようにする)

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



```
1 class MyClass(GeneratedClass):
2     def __init__(self):
3         GeneratedClass.__init__(self)
4
5     def onLoad(self):
6         self.frameManager = ALProxy("ALFrameManager")
7         self.folderName = None
8
9     def onUnload(self):
10        import sys
11        if self.folderName and self.folderName in sys.path:
12            sys.path.remove(self.folderName)
13            self.folderName = None
14
15    def onInput_onStart(self):
16        import sys, os
17        self.folderName =
18        os.path.join(self.frameManager.getBehaviorPath(self.behaviorId), "../lib")
19        if self.folderName not in sys.path:
20            sys.path.append(self.folderName)
21
22    def onInput_onStop(self):
23        self.onUnload() #it is recommended to reuse the clean-up as the box
24        is stopped
25        self.onStopped() #activate the output of the box
```

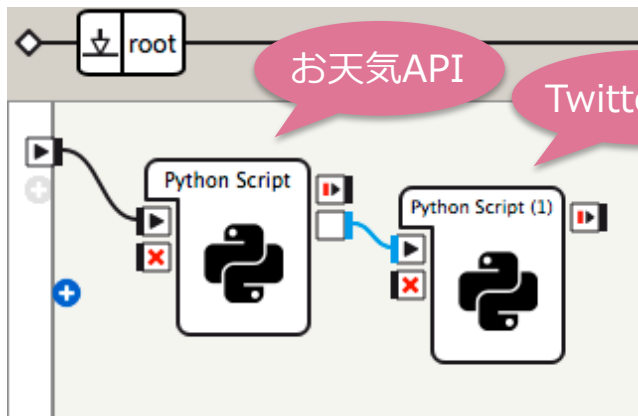
⑧Python Script をダブルクリックし、Boxの中身を変更 (Twitter API の実行)

```
15| def onInput_onStart(self, p):  
    import sys, os  
    self.folderName = os.path.join(self.frameanager.get  
BehaviorPath(self.behaviorId), "../lib")  
    if self.folderName not in sys.path:  
        sys.path.append(self.folderName)
```

指定のキー 4 種類入力
(配布資料に記載)

```
def onInput_onStart(self, p):  
    import sys, os  
    self.folderName = os.path.join(self.frameanager.get  
BehaviorPath(self.behaviorId), "../lib")  
    if self.folderName not in sys.path:  
        sys.path.append(self.folderName)  
  
    from requests_oauthlib import OAuth1Session  
    url = "https://api.twitter.com/1.1/statuses/update.json"  
  
    Consumer_Key = ' ※Consumer Keyを入力 '  
    Consumer_Secret = ' ※Consumer Secretを入力 '  
    Access_Token = ' ※Access Tokenを入力 '  
    Access_Token_Secret = ' ※Access Token Secretを入力 '  
    twitter = OAuth1Session(Consumer_Key,  
                             Consumer_Secret,  
                             Access_Token,  
                             Access_Token_Secret)  
  
    params = {'status':p}  
    twitter.post(url, params = params)  
    self.onStopped()
```

Twitter API でツイート




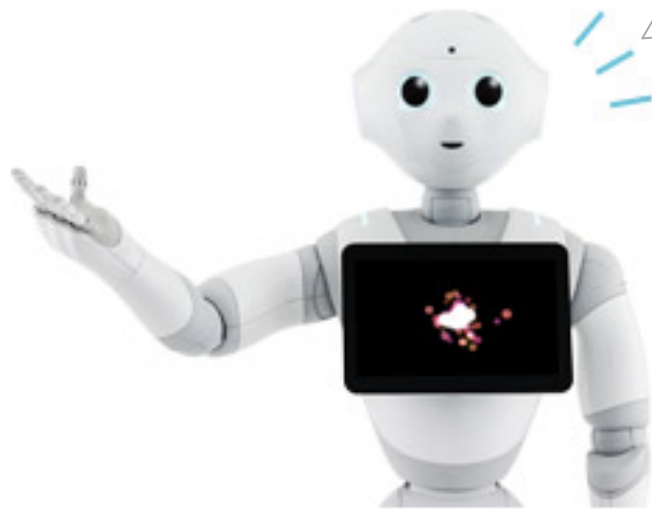
⑨左のようにボックスを配置して完成！

※同じ内容を繰り返しツイートできないので注意！

- ・時間をおいて試す
- ・ツイート内容を変える（例：地域を変える）

おまけ：

- 自分のTwitter アカウントに投稿したい
 - 4種類のキーを別アカウントのものに変更すればOK
 - （参考）Twitterキーの取得の仕方：<https://goo.gl/DeGVGs>
- BoxまわりのTips
 - Box名やアイコンの変更：右クリック→[ボックスを編集]
 - Boxの保存：<https://goo.gl/osbsUt>
 - パラメータボタンで4種キーを入力する：<https://goo.gl/QpfjKI>



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

- ・ 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

- NAO (Pepper) に天気を教えてもらう

<https://goo.gl/BuXIBD>

- Python で Twitter API にアクセス

<https://goo.gl/a3csxK>

- Pepper用Tweetボックスをつくる

<https://goo.gl/QpfjKI>

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。
タッチアンドトライでの質問もお待ちしています。

