

Atelier Akihabara

ワークショップ上級

2017/5/21

Softbank Robotics



軽く自己紹介をしましょう！

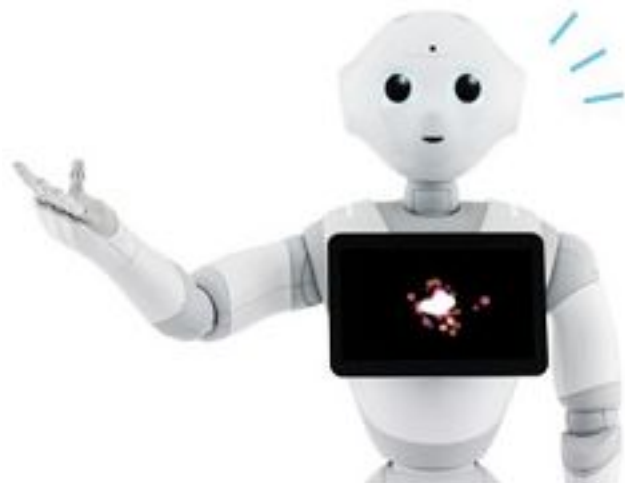
- お名前
- 所属
- プログラミング経験や本日の意気込み
- html, JSに関する経験 など

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

目次

1. QiMessaging JavaScriptとは
2. 簡単なHTMLをタブレットに表示
3. 簡単なJavaScriptの実行
4. androidからNAOqiに指示を送る
5. NAOqiからandroidに指示を送る

タブレットとNaoqiの相互のやり取りを学ぶことが目標です



配布物

- 本プレゼンテーション資料
- プロジェクト x 4

プロジェクト名	内容	目的
show-app	htmlページの表示	htmlページの表示
simple-js	ボタンを押すと発話	html+JavaScriptの実行
memory-from-tablet	ボタンを押すと発話(3種類)	タブレットの情報を本体で利用 (android → NAOqi)
memory-to-tablet	人にIDを付けてタブレットに表示	本体の情報をタブレットで利用 (NAOqi → android)



Pepperの
タブレットって？

Pepperのタブレット

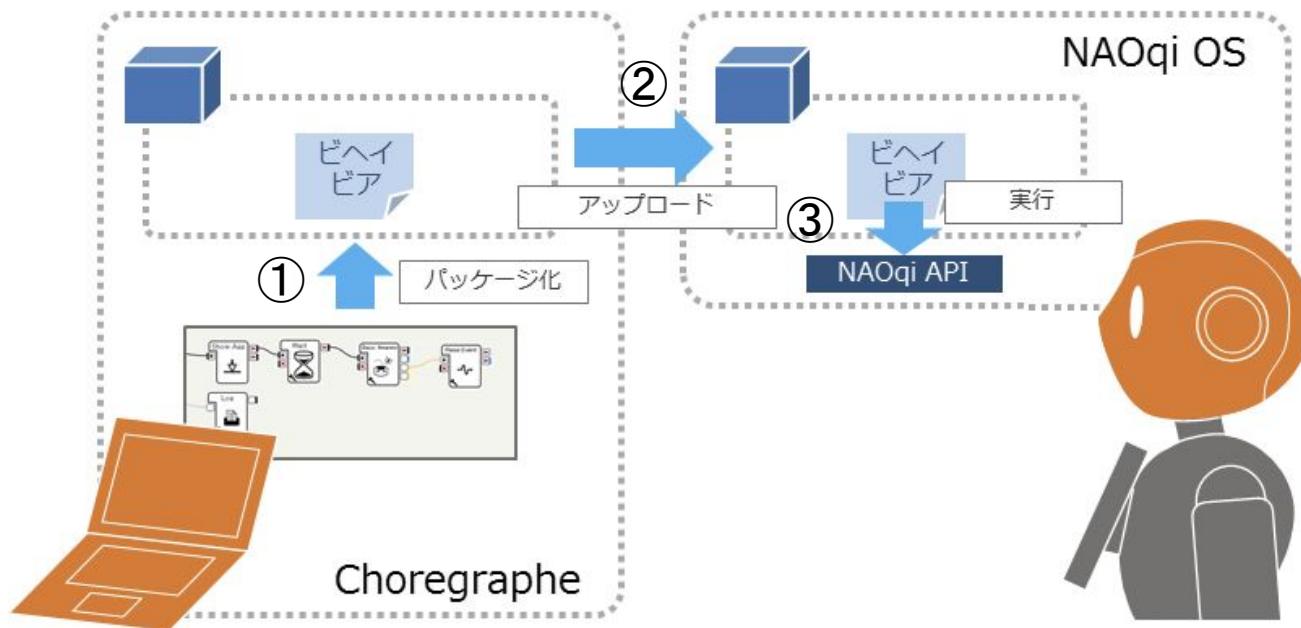


PepperのOS(NAOqi OS)とは独立したOS(Android OS)が動作している

1. 画像・動画などをうつす
2. JavaScriptなどを含むHTMLコンテンツを実行する
3. タッチイベントを取得することが可能

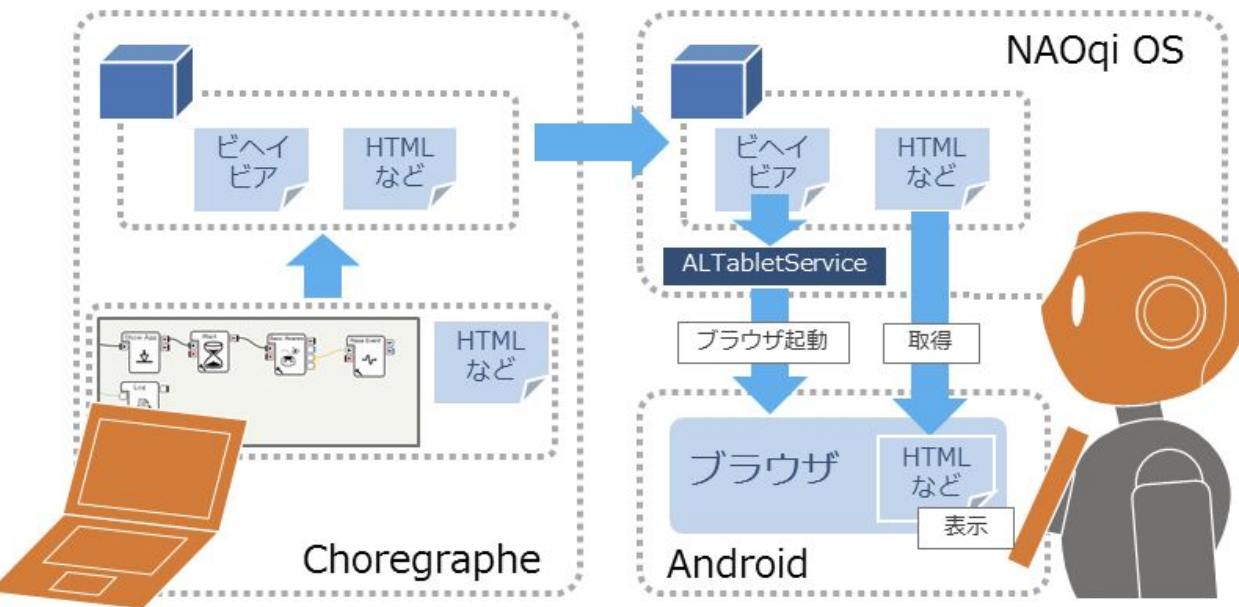
よりリッチなアプリを作るためにはタブレットの開発が必須！

NAOqiでのアプリケーション実行



- ①Choregraphe上で作成したプログラムをパッケージング
- ↓
- ②Pepperにアップロード
- ↓
- ③それぞれのボックスがNAOqi APIにアクセス
- ↓
- ④アプリを実行

タブレットでの画像や動画の表示

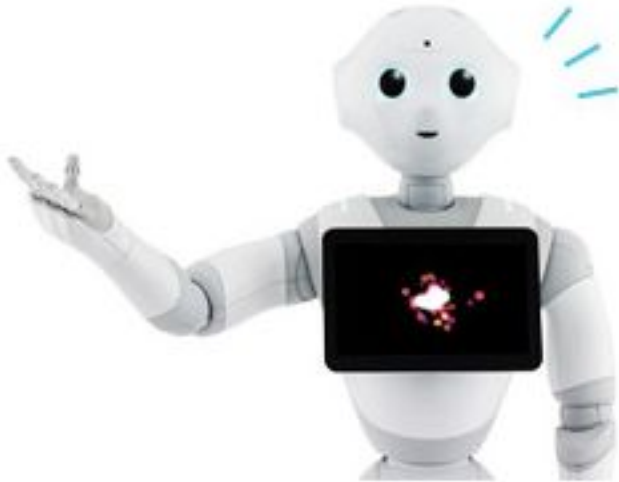


- ・htmlフォルダが読み込まれる
- ↓
- ・NAOqiが指示を送りAndroidが動く

※Choregrapheから最初にデータを送るのはNAOqi OS

- ・タブレットで画像やHTMLを読み込むときはALTabletServiceというAPIを通じて指示を送る

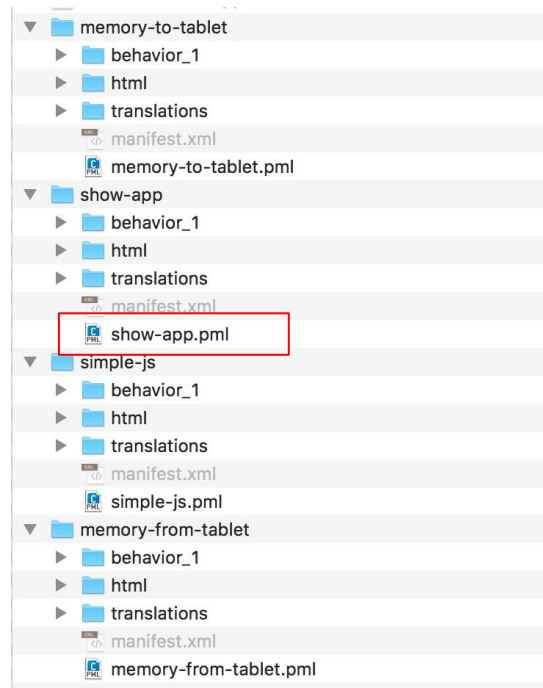
簡単なHTMLを
タブレットに表示
show-appプロジェ
クト



① ファイル > プロジェクトを開く



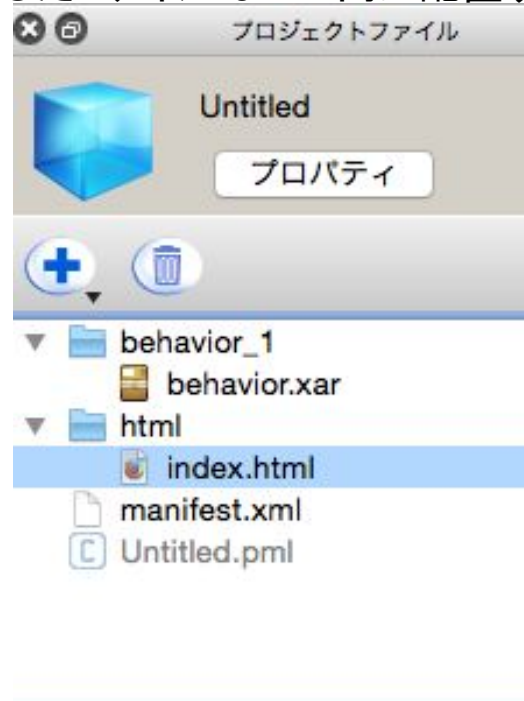
② ワークショップhard_1_2.5フォルダ内 show-app > show-app.pmlを選択

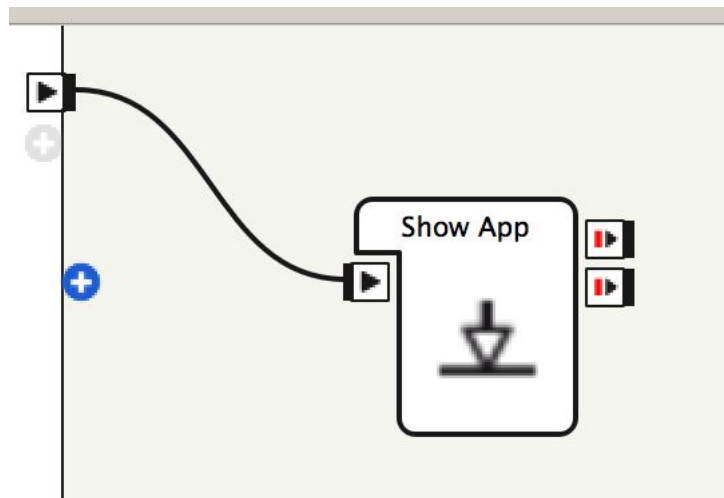


① +ボタン>[新規フォルダ...]を選択し、
名前を「html」として作成

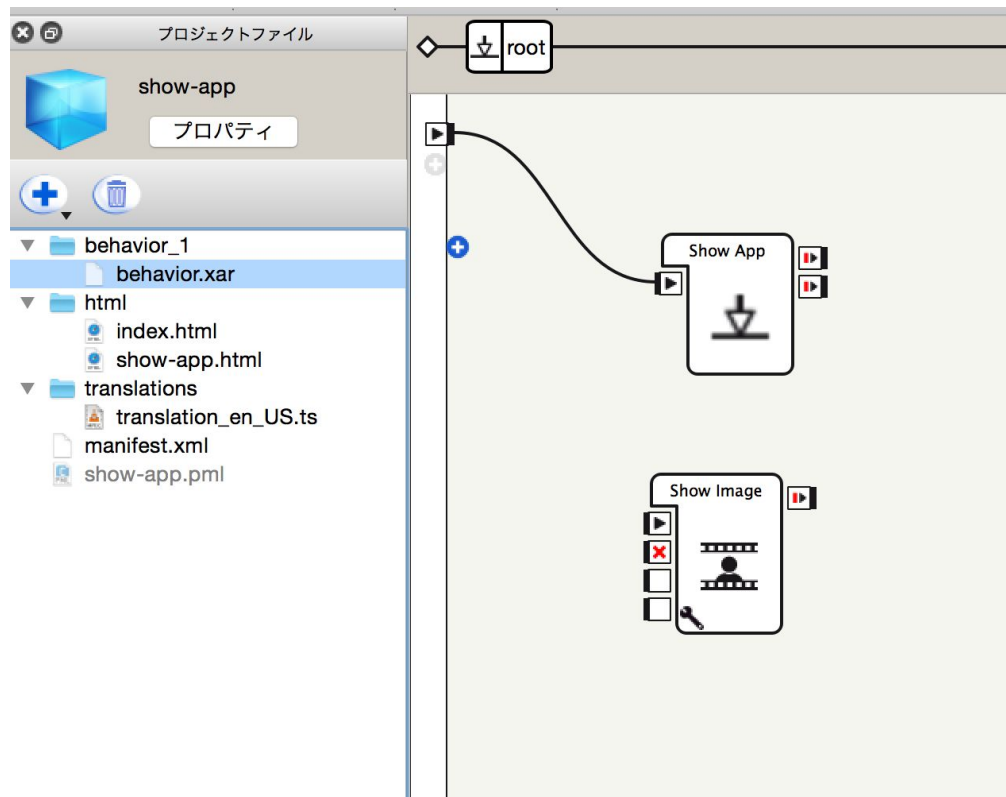


② +ボタン>[ファイルをインポート...]を選択し、
index.htmlをインポートする。
インポートしたファイルはhtml内に配置する。



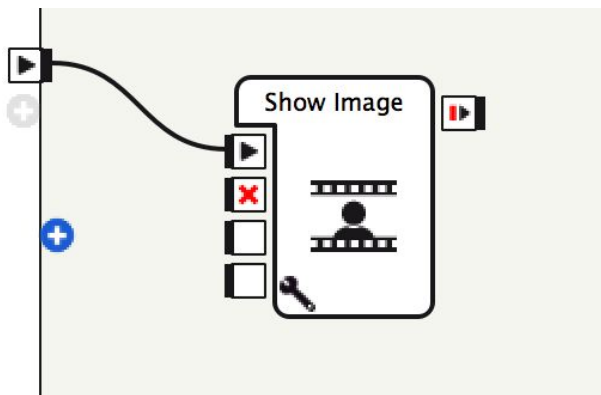


Show App BOXは
「htmlフォルダのindex.html」
を表示させるボックス



パターン2

「index.html」以外の名前を使いたいときはshow imageボックスを修正



htmlファイルを保存する際の文字コードは**UTF-8**

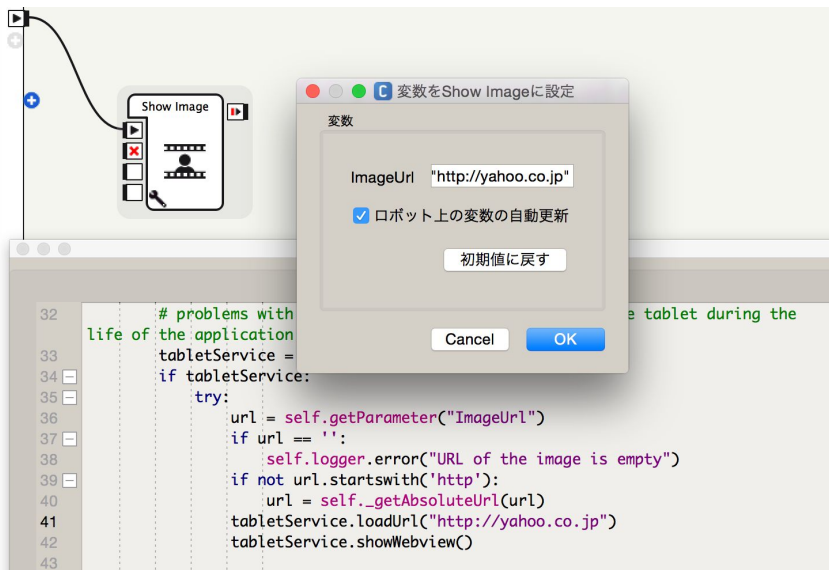
```
37|tabletService.showImage(url)
```

#上記のようになっている37行目を
以下のように上書きする

```
37|tabletService.showWebview(url)
```

パターン2 の応用

タブレットでブラウジングしたいときはshow imageを以下のように使う



37|tabletService.showImage(url)

#上記のようにになっている37行目を
以下のように上書きする

37|tabletService.showWebView("http://yahoo.co.jp")

- ・imageUrlの中に""で囲んでURLを記載
- ・41行目にも""で囲んでURLを記載

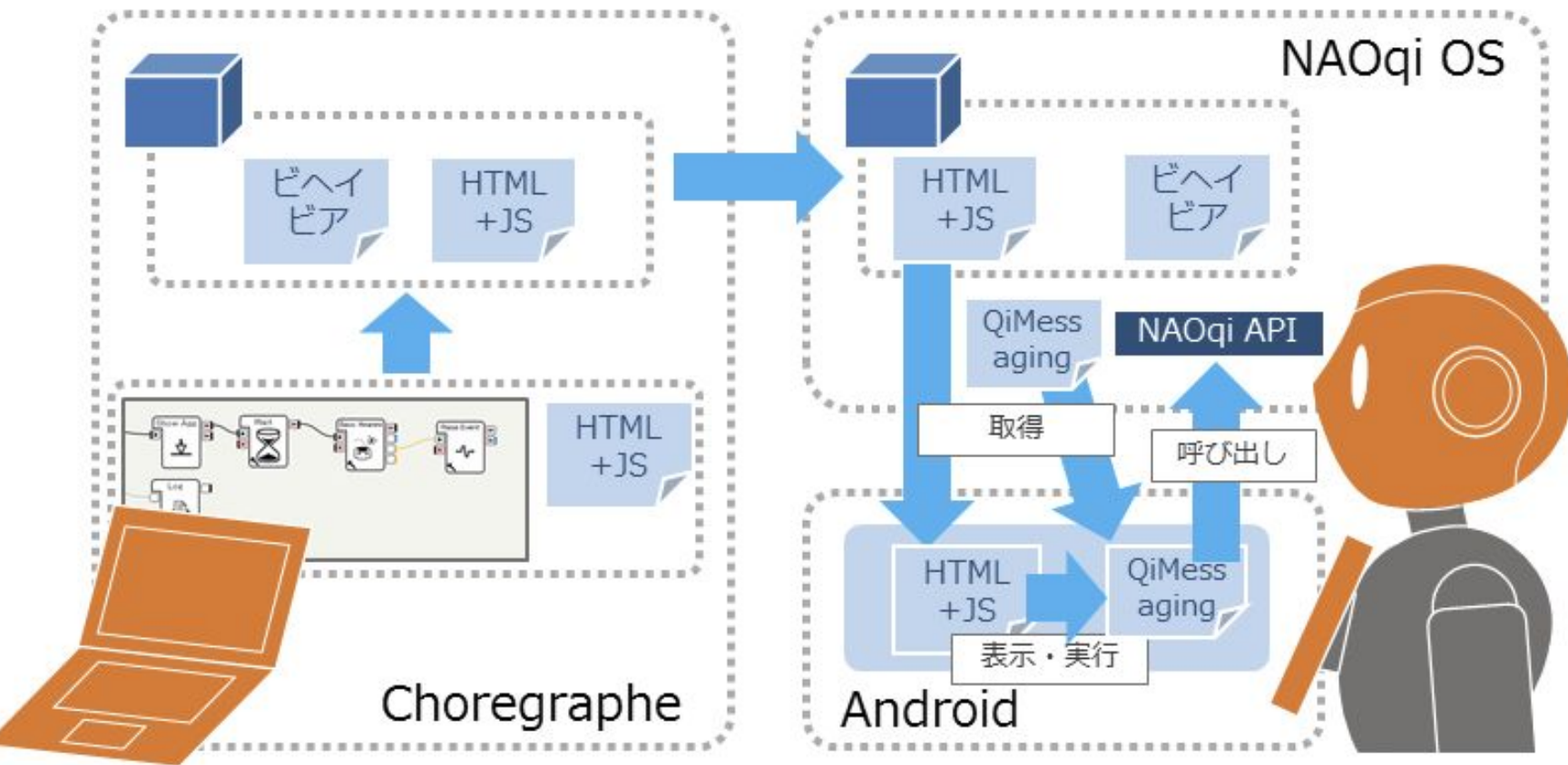
簡単な
JavaScriptの実行
「simple-js」



QiMessaging JavaScript とは



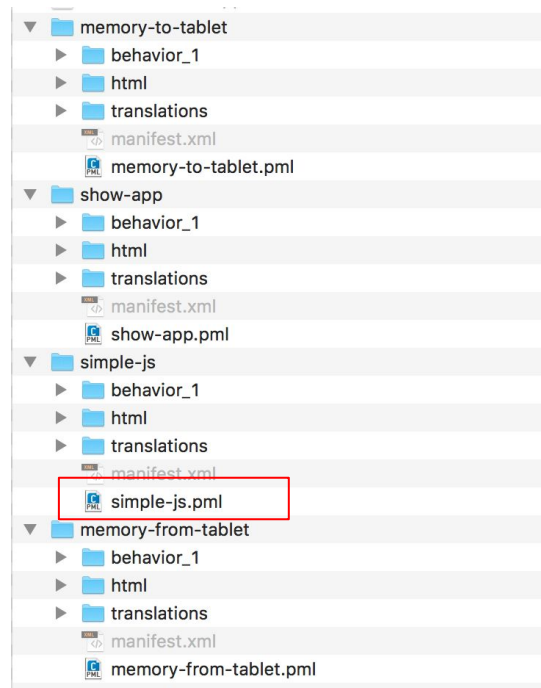
QiMessaging JavaScript



① ファイル > プロジェクトを開く



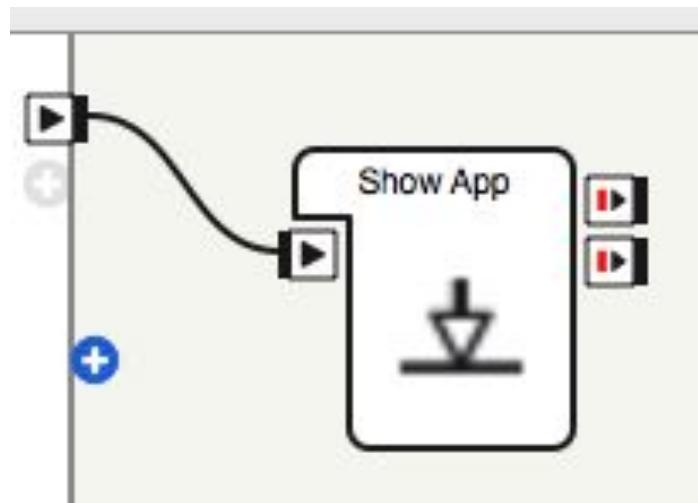
② ワークショップhard_1_2.5フォルダ内 show-app > simple-js.pmlを選択



- ① +ボタン>[ファイルをインポート...]を選択し、
index.htmlをインポートする。
html内に配置する。



- ② Show Appボックスを用
意する



ボタンが記述されたHTMLが表示される

- qimessaging.jsの読み込み

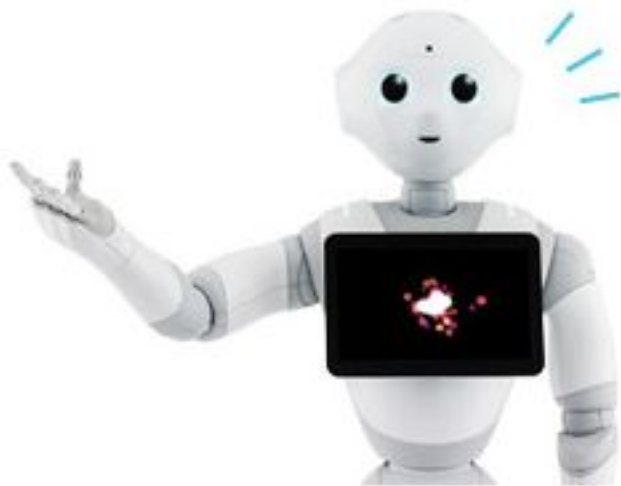
```
7|<script src="/libs/qimessaging/2/qimessaging.js"></script>
```

- QiSessionオブジェクトの作成

```
ver session = new QiSession();
```

- QiSessionからのALTextToSpeech取得、tts.say呼び出し

```
15|session.service("ALTextToSpeech").done(function (tts) {  
16|    tts.say("こんにちは");  
17|});
```



android→NAOqi
「memory-from-tablet」

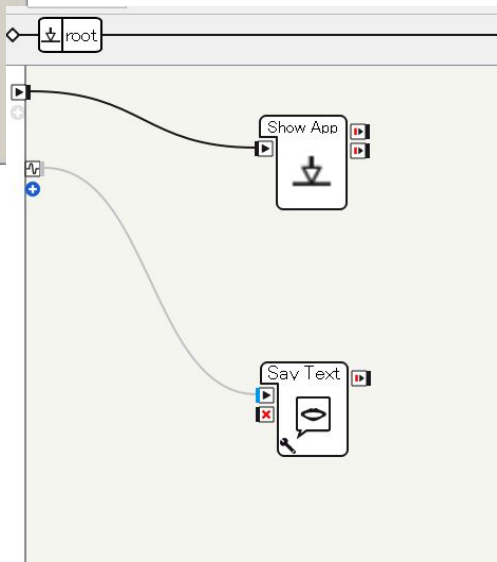
タブレット→アプリケーション ①

memory-from-tablet

① +ボタンから[ファイルをインポート...]を選択、
index.htmlをインポートする。
html内に配置する



② ALTabletService/message という名前の
メモリーキーを追加し、Say Textボックスとつなぐ



- qimessaging.jsの読み込み

```
7|<script src="/libs/qimessaging/2/qimessaging.js"></script>
```

- QiSessionオブジェクトの初期化

```
9|var session;
10|QiSession( function(s){
11|session = s;
12|});
```

- ボタンが押された際、sessionオブジェクトに対して service関数を通じて
ALMemoryへのアクセス用オブジェクト要求

"イベント発生"という文字列値が keyとして入った

"PepperQiMessaging/fromtablet" イベントを Memoryに発生させる

```
15|session.service("ALMemory").then(function (ALMemory) {
16|console.log("ALMemory取得成功");
17|ALMemory.raiseEvent("PepperQiMessaging/fromtablet", "イベント発生");
18|    });
```

- 「say from JS」ボタンクリックからserviceを通じて直接発話

```
22|session.service("ALTextToSpeech").then(function (tts) {  
23|  tts.say("APIを使って発話");  
24| });
```

- 「hello world」ボタンクリックからタグ要素内の引数で指定文字列を発話

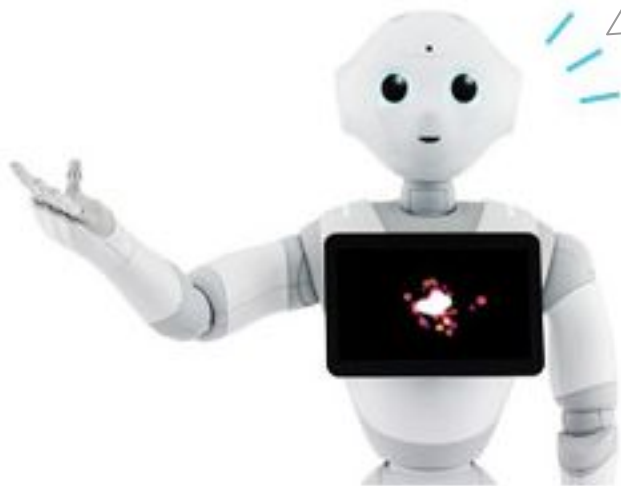
```
27|function sayWithParam(text) {  
28|  session.service("ALTextToSpeech").then(function (tts) {  
29|    tts.say(text);  
30|  });
```

```
43|<button style="font-size: 6em" type="button"  
    onclick="sayWithParam('こんにちは');">こんにちは</button>
```

※引数とは...

プログラム中で、関数に値を渡すパラメータ

タブレットの内容を
PCで確認



ディスプレイ表示のデバック

QiMessaging (raise
memory event)

イベント発生

Javaスクリプトで発話

こんにちは

<http://<pepperのIPアドレス>/apps/アプリのID>
にPCのブラウザからアクセス

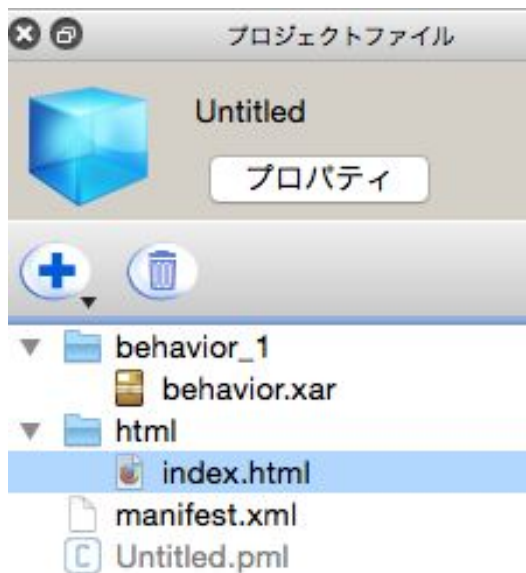
memory-from-tabletを表示させる場合
<http://atelier○○.local/apps/.lastUploadedChoregrapheBehavior/>
とブラウザのURLに入力。

※レイアウトがPepperのディスプレイと異なることがある

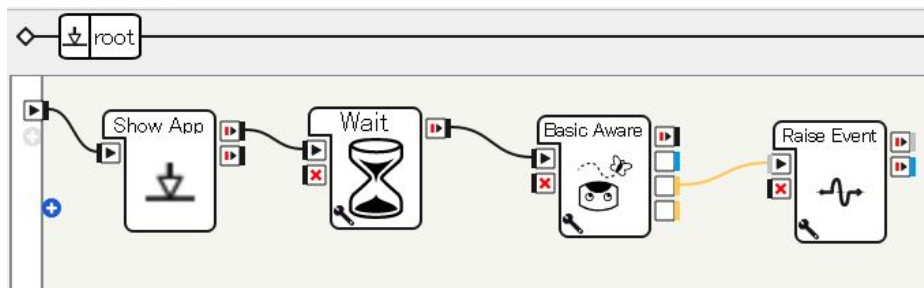
NAOqi→android
「memory-to-tablet」



① +ボタンからファイルをインポート...を選択
index.htmlをインポートする
インポートしたファイルはhtml内に配置する



② 下図のようにボックスを配置する



③ イベントを発火させるため、
Raise Eventボックスの変数のkeyに
「PepperQiMessaging/totablet」を設定する



- qimessaging.jsの読み込み

```
7|<script src="/libs/qimessaging/2/qimessaging.js"></script>
```

- QiSessionオブジェクトの作成

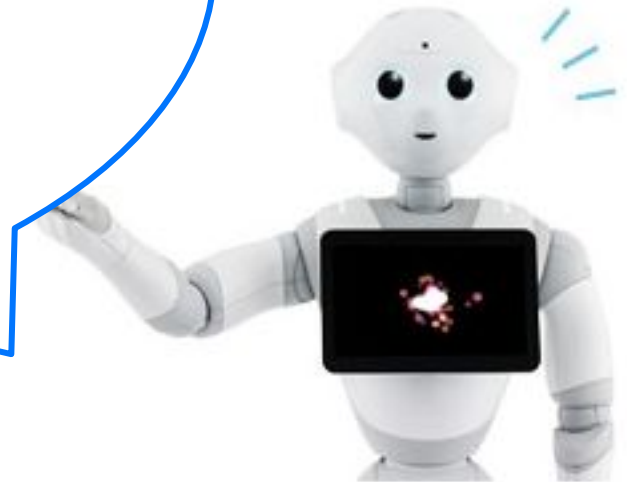
```
9|var session;  
10|QiSession( function(s){  
11|session = s;  
12|});
```

- ALMemory.subscriber関数によるイベント監視

```
26|ALMemory.subscriber("PepperQiMessaging/totablet").then(function(subscriber) {  
27|    subscriber.signal.connect(toTabletHandler);  
28|    });
```

プロジェクト名	内容	目的
show-app	htmlページの表示	htmlページの表示
simple-js	ボタンを押すと発話	html+JavaScriptの実行
memory- from-tablet	ボタンを押すと発話(3種類)	タブレットの情報を本体で利用 (android → NAOqi)
memory- to-tablet	人にIDを付けてタブレットに表示	本体の情報をタブレットで利用 (NAOqi → android)

おつかれさまでした！
これにてWS上級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
お帰りの際はアンケートの記入に
ご協力ください



<https://bit.ly/pepperatelier>

