

# RoboCup@Home Education

## ONLINE CHALLENGE 2020

### Online Classroom Standard Platform

## 04 Dialogues

RoboCup@Home Education | 2020.05.07

**RoboCup@Home**  
**EDUCATION**

 **SoftBank**  
Robotics

# Online Challenge 2020: Online Classroom SP

## 04 Dialogues

Speakers: Luca locchi, Jeffrey Tan, SoftBank Robotics

Time: **May 7, 2020 (Thu) 19:00 - 20:00 (GMT+8)**

Zoom: <https://cernet.zoom.com.cn/j/66694449441> | PW: robocup

SP Pepper 2.9 Online Support Session [English]

Every Monday 14:30~ (GMT+2) | May 4 ~ May 25, 2020

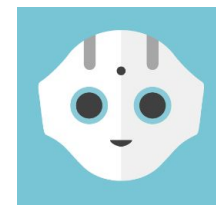
<https://meet.google.com/pvq-sjkq-efr>

Online Classroom:

<https://www.robocupathomeedu.org/learn/online-classroom/online-challenge-2020>

\*\* Privacy reminder: Video will be recorded and published online.

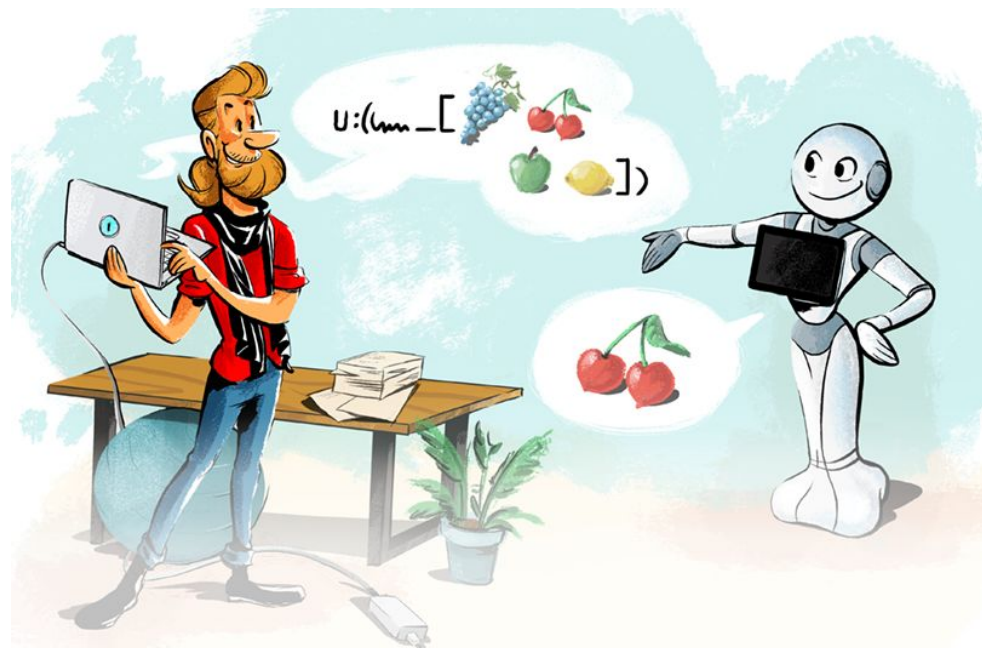
# QiChat



High-level specification  
of dialogues

QiChat script language

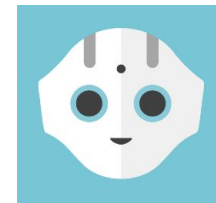
Chat actions



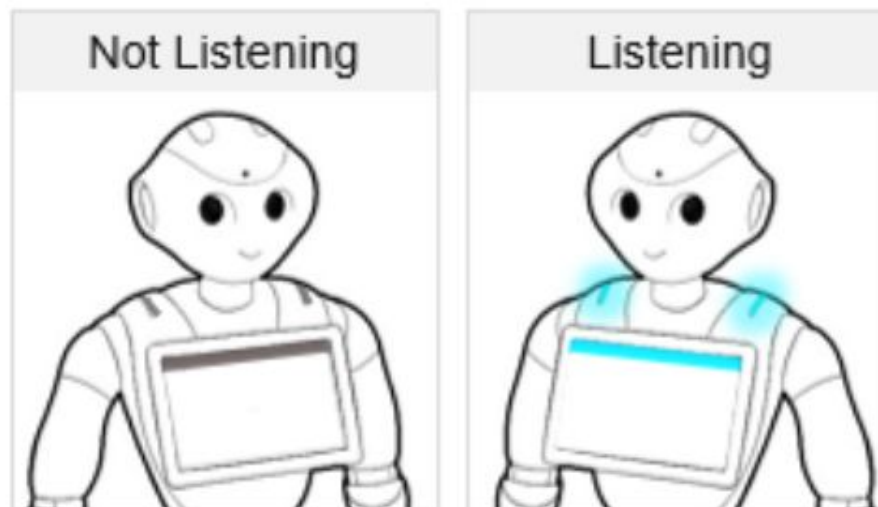
<https://developer.softbankrobotics.com/pepper-qisdsk/api/conversation>

[https://developer.softbankrobotics.com/pepper-qisdsk/api/conversation/  
qichat-language](https://developer.softbankrobotics.com/pepper-qisdsk/api/conversation/qichat-language)

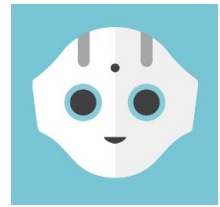
# Listening states



When Speech Recognition is active, light blue **SpeechBar** is enabled

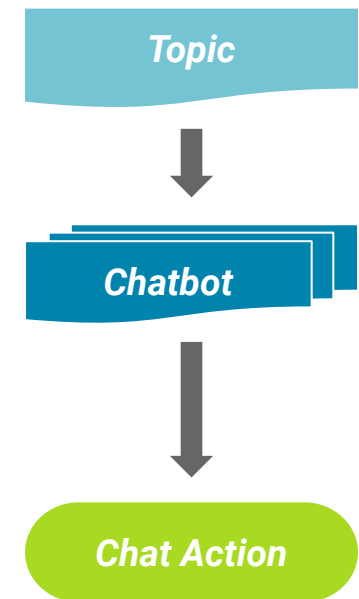


# Conversations

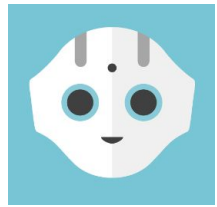


## Programming steps

1. Create a Topic
2. Define a Chatbot
3. Define and run a Chat action



# Chat topic



File -> New -> Chat Topic

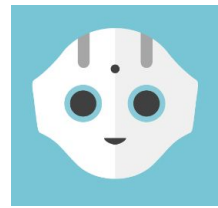
**topic:** ~greetings()

# Defining extra concepts out of words or group of words

**concept:**(hello) [ hello hi hey "good morning" greetings ]

# Replying to speech

**u:**(~hello) ~hello



# Chat action

*// Create a chat topic*

```
Topic topic = TopicBuilder.with(qiContext) // Create the builder  
    .withResource(R.raw.greetings) // Set the topic resource  
    .build(); // Build the topic.
```

*// Create a new QiChatbot.*

```
QiChatbot qiChatbot = QiChatbotBuilder.with(qiContext)  
    .withTopic(topic)  
    .build();
```

*// Create a new Chat action.*

```
Chat chatAction = ChatBuilder.with(qiContext)  
    .withChatbot(qiChatbot)  
    .build();
```

*// Run the Chat action asynchronously.*

```
chatAction.async().run();
```



# QiChat language - Concepts

Words associated to semantic meanings

**concept:**(hello) [ hello hi hey "good morning" greetings ]

**concept:**(bring) [ bring get take ]

**concept:**(bringme) [ "{please} ~bring {me} {some} {a}" ]

Name: ( ) round brackets

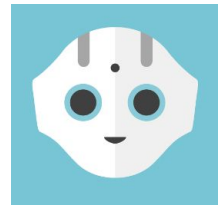
Choice: [ ] list of words square brackets

Optional words { } curly brackets

Concept use ~ tilde symbol before concept name



# QiChat language - Rules and sub-rules



Robot reply to human input

**u:( *human input* ) *robot output***

**u:(hi)** hello

**u:(~hello)** ~hello

**u:(~hello)** Do you like animals?

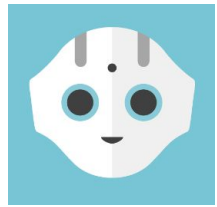
**u1:(yes)** Which is your preferite one?

**u2:(dog)** I also like dogs

**u2:(cat)** Cats are very nice

**u1:(no)** Oh. Maybe you prefer something else

# QiChat language - Functions



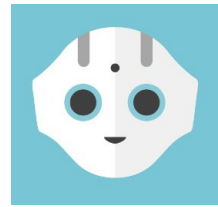
**^rand** [hello hi hey "good morning" greetings]

**^first** [hello hi hey "good morning" greetings]

**^nextProposal**

**^stayInScope**

# QiChat language - Variables



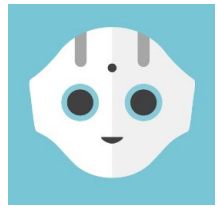
## Variables

**u:(~hello)** Hi. What's your name?

**u1:(~\*)** Hi **\$1**, nice to meet you **\$Name=\$1**

**u:(~bye)** **^first** [ "Bye bye **\$Name**" "Bye bye" ]

# QiChat language - Proposals



**u:(~hello) ~hello ^nextProposal**

**proposal:** How are you?

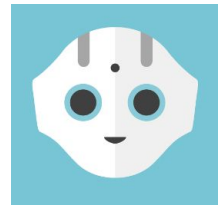
**u1:(~feel-good)** Great.

**u1:(~feel-bad)** Oh. I am sorry.

**proposal:** What is your name?

**u1:(\_\*)** Hi **\$1**, Nice to meet you **\$Name=\$1**

# QiChat & Java - ChatVariable callbacks



*// Get a reference to a GUI element*

```
TextView textView = (TextView)findViewById(R.id.textView1);
```

*// Set up a listener for a chat variable*

```
QiChatVariable nameVariable = qiChatbot.variable("Name");
```

```
nameVariable.addOnValueChangedListener(  
    currentValue -> {  
        Log.i(TAG, "Chat var Name: " + currentValue);  
        textView.setText("Hello " + currentValue);  
    }  
);
```

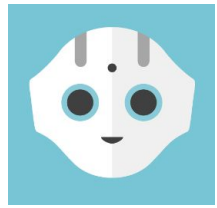
# Basic Chatbot App



## BasicChatbot

[https://github.com/robocupathomeedu/rc-home-edu-learn-pepper/  
blob/master/test\\_activities/](https://github.com/robocupathomeedu/rc-home-edu-learn-pepper/blob/master/test_activities/)

# Code and documentation repositories



<https://github.com/robocupathomeedu/rc-home-edu-learn-pepper>

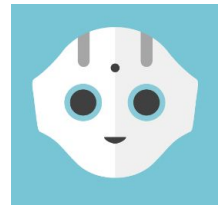
<https://github.com/aldebaran/qisdk-tutorials>

Code examples

Documentation

Issues (Q&A)

# Assignment 2 - AnswersAboutMyTeam



Develop a Pepper app: **AnswersAboutMyTeam** implementing the following behavior

The tablet screen shows some information about your team and a Start button.

When the app starts, the robot greets people. When the Start button is pushed, Pepper starts answering questions about your team using a chat topic.

Consider questions in the team registration form.

Use github classroom to develop the assignment.





# Q&A session and docs

RoboCup@Homeedu OnLine 2020  
Standard Platform (Pepper Android SDK)

Monday 2:30-3:30 pm (GMT+2 / CEST)

Google Meet - <https://meet.google.com/pvq-sjkq-efr>

## Q&A Book

[https://docs.google.com/document/d/1CU9AyS2tMRBGkvX8jZop\\_PZn-mOLFJ4sFRGKE8TaUXM](https://docs.google.com/document/d/1CU9AyS2tMRBGkvX8jZop_PZn-mOLFJ4sFRGKE8TaUXM)



# Next lectures

01 Introduction

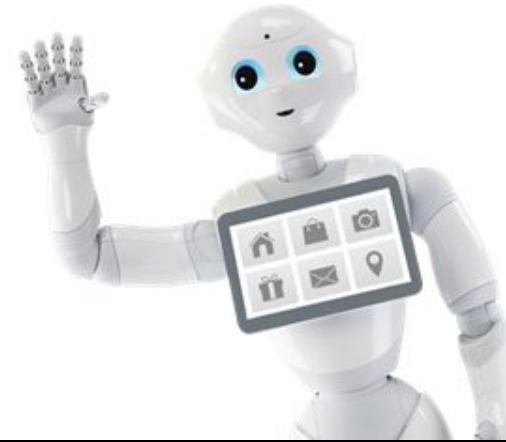
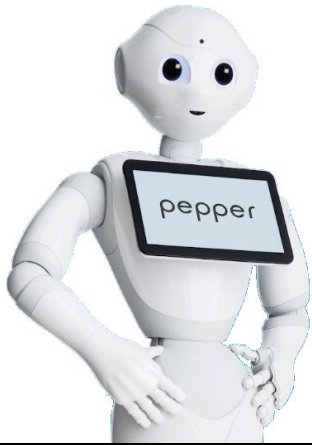
02 Pepper and Software Installation

03 Apps Programming

04 Dialogues

**05 Examples**

**06 Advanced Programming**



# RoboCup@Home Education

## ONLINE CHALLENGE 2020

### Online Classroom Standard Platform

Web: <https://www.robocupathomeedu.org/challenges/robocuphome-education-online-challenge-2020>

Online Classroom: <https://www.robocupathomeedu.org/learn/online-classroom/online-challenge-2020>

Online Entry Form: <https://forms.gle/UBREeC1xTCVQ9wr78>

Online Entry Form (backup): <https://www.wjx.cn/jq/72082120.aspx>

Contact: [oc@robocupathomeedu.org](mailto:oc@robocupathomeedu.org)

**RoboCup@Home**  
**EDUCATION**

 **SoftBank**  
Robotics