

1. Getting started

In this project, your smartphone will represent the eyes of your digital assistant. This works best if the smartphone is placed crosswise!

To connect your smartphone to Snap! proceed as follows:

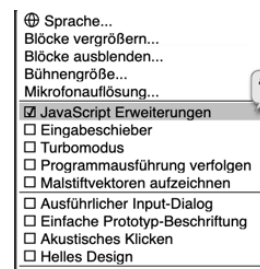
- a) On your smartphone, open the URL:
<https://it2school.imp.fu-berlin.de/>
 You can also scan the QR code on the right!
- b) In order for your digital assistant on your smartphone and your Snap! project to be associated with each other, a unique "room name" must be assigned. To do this, enter the name of your assistant followed by a combination of numbers and write it down here:



-
- c) Open the Snap! template on your computer:
<https://snap.berkeley.edu/snap/snap.html#present:Username=neuner&ProjectName=it2school-Digitaler-Assistent&editMode&noRun>

Save the project right away under your assistants name in the Snap! cloud or on your computer.

- d) Enable **JavaScript extensions** by clicking on the cogwheel in the upper left corner and checking the corresponding check mark. Then press "i" to start the script to initialize the assistant and follow the instructions. By pressing "t" (test) the connection can be tested and renewed later.



- e) Try out whether your assistant now receives instructions by clicking on the blocks `set emotions on checkleft` and `set emotions on reallysad`. You can now also try other emotions from the drop-down menu! If everything works, you will see the selected emotions on your smartphone.

- f) Now try the blocks `say as digital assistant Hi` to make your assistant speak..

Tip: If the connection with the smartphone is interrupted, press "t" to reconnect and test or simply reload the page on the smartphone.

2. Familiarize with the project

Our digital assistant already has the functionality of a classifier: the AI model can be trained to assign the categories "positive" or "negative" to texts, for example.

In the script area you can find the following script:




- a) Create and train the AI model with the given positive and negative sentences by clicking on them. It is clear that the classifier assigns the sentences used for training according to the category specified in the process (feel free to try it yourself if it is true!). But how does it handle unknown sentences?

Test the classifier by calling it with the following two given sentences and one sentence of your own:

determine category for <input type="text"/>	resulting category	actual category
I am doing fantastic!	<i>positive</i>	<i>positive</i>
I feel absolutely great!		

- b) Apparently, the AI model does not yet provide correct results in all cases. Extend the training data of the classifier with more sentences and train again. To do this, you have to adapt the script from the beginning of this task and run it again. Improve your AI model until the above sentences are recognized correctly without using the concrete sentences as training data.

Note: If the AI model cannot determine a category, the classifier returns "undetermined".

Tip: You can add new sentences to  by clicking t
black arrow on the right..

3. Start the personal assistant

Now you have to combine the two functionalities from task 1 and task 2: Your assistant has learned to determine the mood of texts and should now apply this ability to texts spoken by you and respond with the appropriate eye expression and a short answer.

In order for your assistant to listen, you must briefly tap the display before and after speaking.

Start the script for the digital assistant with the green flag. Now tap on the display and speak to your digital assistant, for example with a sentence like "I'm fine" - he should now show a friendly face and answer you.

Note: Voice input is not available with all smartphones or browsers. Try a different system if necessary.



Tip: You can display the result of the speech recognition and the recognized mood on the Snap! stage by placing a check mark in front of the corresponding variables.

4. Extend the project

Your digital assistant still answers quite briefly. Now expand the project so that you can have a small conversation.

5. Create your own assistant

This was just one of many ways to extend the program behind your personal digital assistant. Of course, the classifier can not only assign moods to the categories positive and negative, but also, if trained accordingly, identify other categories, such as happy, sad, melancholic or silly. Besides moods, any other categorizable fact can be trained. You can find some examples in the following table.

Subject	Possible categories
Mood	happy, sad, melancholic, silly, ...
Illness	covid, hay fever, gastrointestinal, ...
Music	pop, rock, rap, classical, ...
User intent	Play music, start timer, tell joke, ...

An example dialog for an assistant that recognizes diseases could look like this:

Assistant: "What symptoms do you have?"

I had to throw up and I have a stomachache.

Assistant: "Oh dear, then you must be suffering from a stomach bug. I can recommend chamomile tea and plenty of fluids."

Background: How does it work?

In a digital assistant, not one but several AI systems are used, each of which has a specific functionality (e.g. speech recognition, speech synthesis and intent recognition via classification). You will learn how this can look like in module AI-B2.

The digital assistant uses the Web Speech API for speech recognition and speech synthesis, which is supported by all common browsers. Depending on the browser, the behind-the-scenes implementation may vary. Central to this module is the creation of a classifier that is trained with supervised learning (see module AI-B3). There are a variety of ways to implement a classifier. Here, the compact package `ml-classify-text-js` (MIT license) is used. It is learned which words are used in the sentences of a category. An unknown sentence is compared to all sentences used in the training in terms of cosine similarity. The unknown sentence is then assigned the class of the most similar training sentence. The procedure is similar to the supervised k-nearest neighbor procedure.