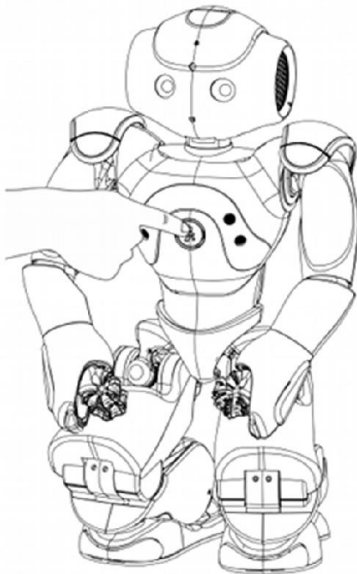
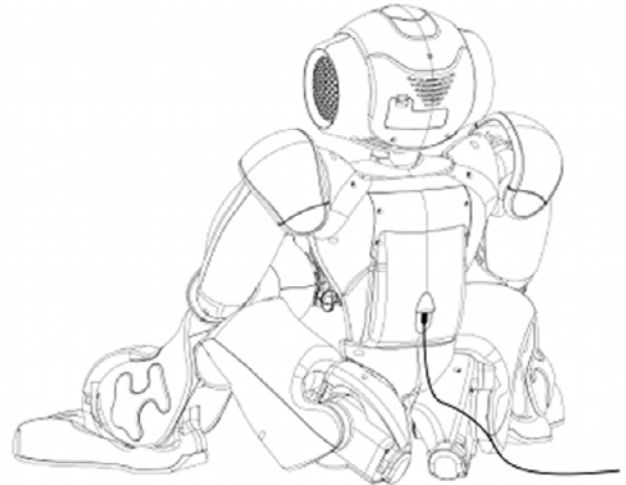


Preparing to Use the NAO

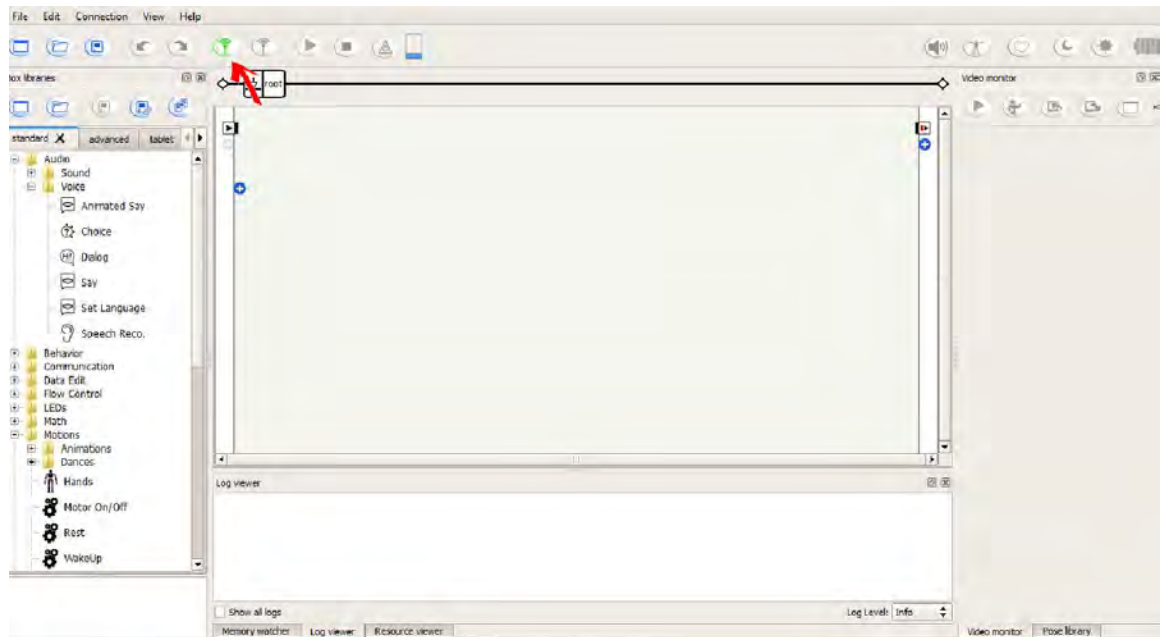
Before we begin to use the NAO, we will learn how to charge, set up, and handle the robot safely. In our figures, **red** arrows mean single-click, **blue** arrows mean click-and-drag, and **green** arrows mean double-click.

1. When the NAO is not moving, plug the charging cable into the robot's back (see picture to right). This will ensure that the robot remains charged for your continued use. If the robot is charging, the charger's light will turn red.
2. Before turning on the NAO, make sure it is in a stable position. The pose shown in the image below, with the back upright and both feet flat on the floor, is recommended. Do not place the NAO on a table where it can fall and damage itself.

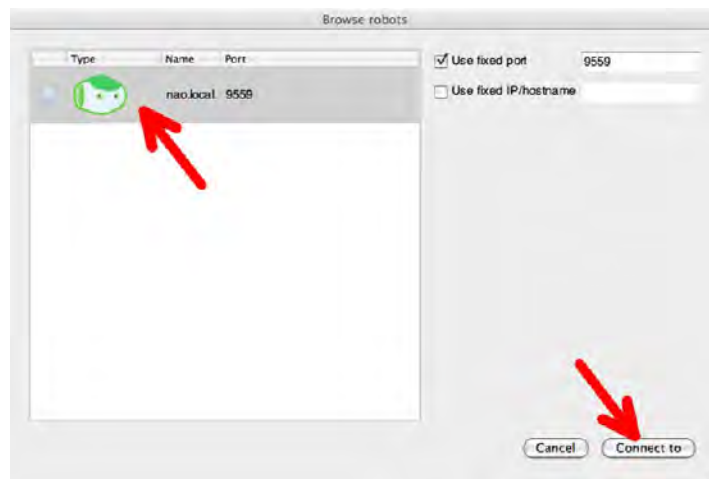


3. Turn on the NAO by pressing the chest button. The NAO's lights will turn on. The NAO can take up to several minutes to boot, and it will make a sound when finished.

4. Now, you will connect to the NAO using Choregraphe. First, start Choregraphe on your computer. The following window will appear. Click the indicated “Connect” button, which is green and looks like a wireless signal, to connect to the NAO.



5. The window below will appear, with your robot listed on the table to the left. Select your robot in the list, and click the “Connect To” button.

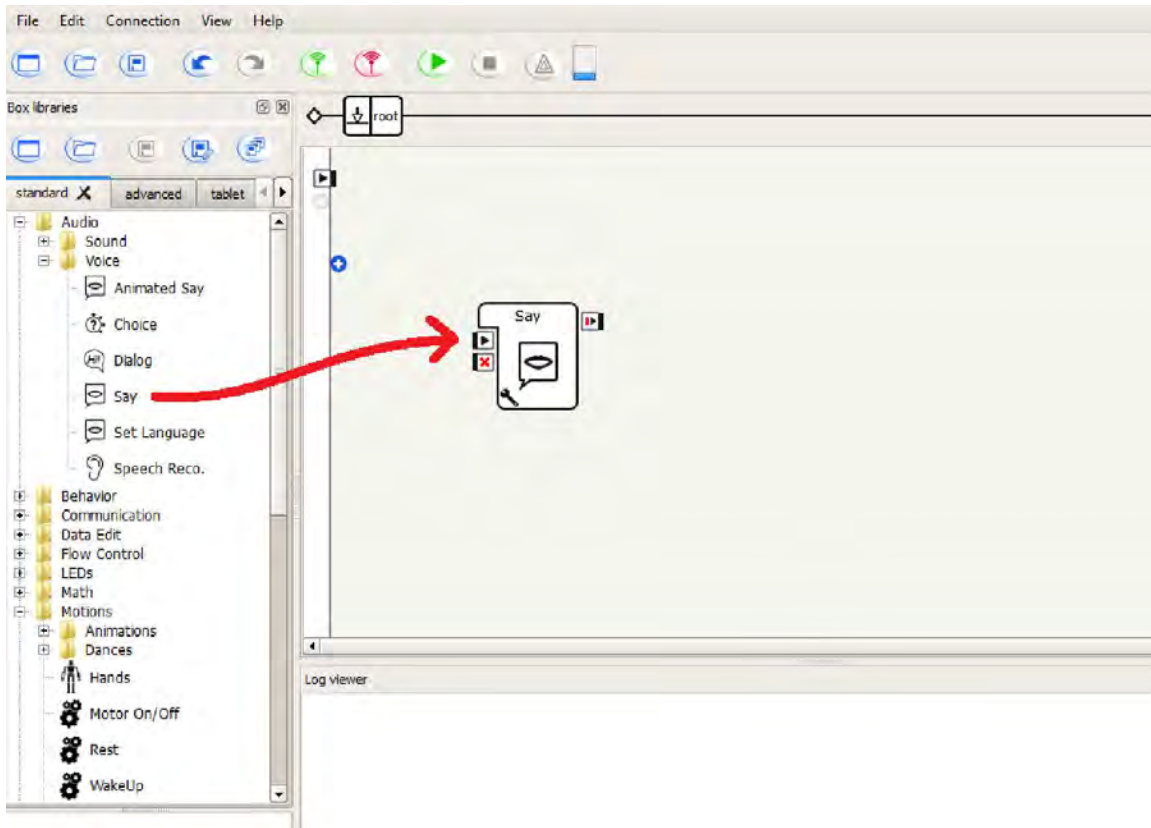


6. Now you should be connected to the NAO. Try moving some of the NAO’s body parts by hand. The window to the right in Choregraphe will update to reflect the changes in the NAO’s joint positions.

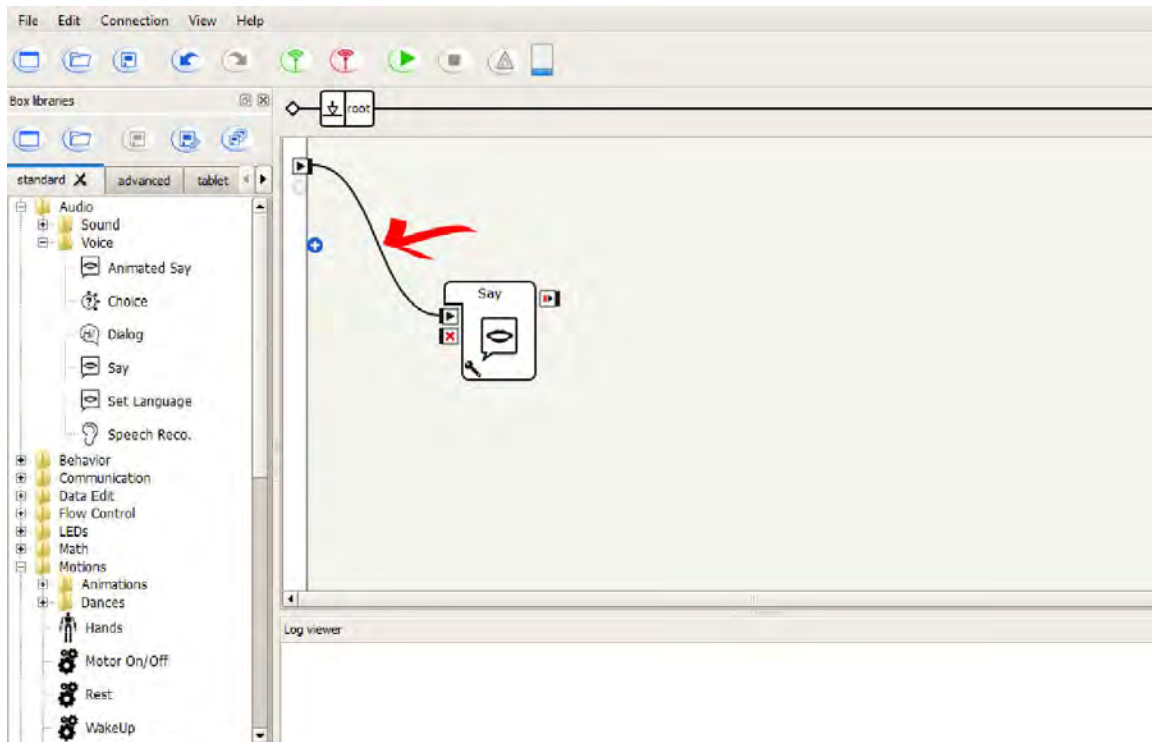
Basic Task: Hello World!

Next, we will program the NAO to say “Hello.” The NAO uses a text-to-speech engine to convert text into sound, which is output through the speakers.

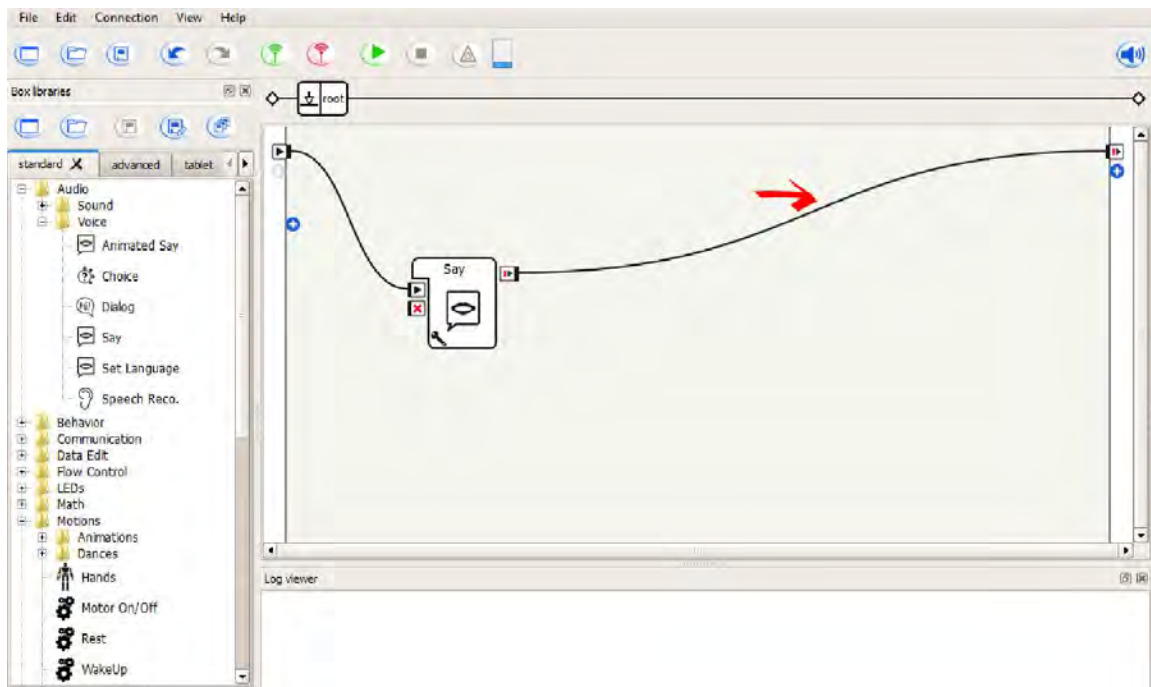
1. In Choregraphe, look at the “Box List” to the left. Navigate to Audio -> Voice, then drag and drop a “Say” box to the central area. This central area is called the workspace, and contains the commands that the robot executes.



2. To execute the “Say” box, we must connect it into the program “flow”. Click on the small arrow to the left of the workspace, and drag a line to the arrow on the left of the “Say” box. The boxes will be executed sequentially in the order that they are connected. The last box should connect to global stop [x] on the right hand side of the work area.



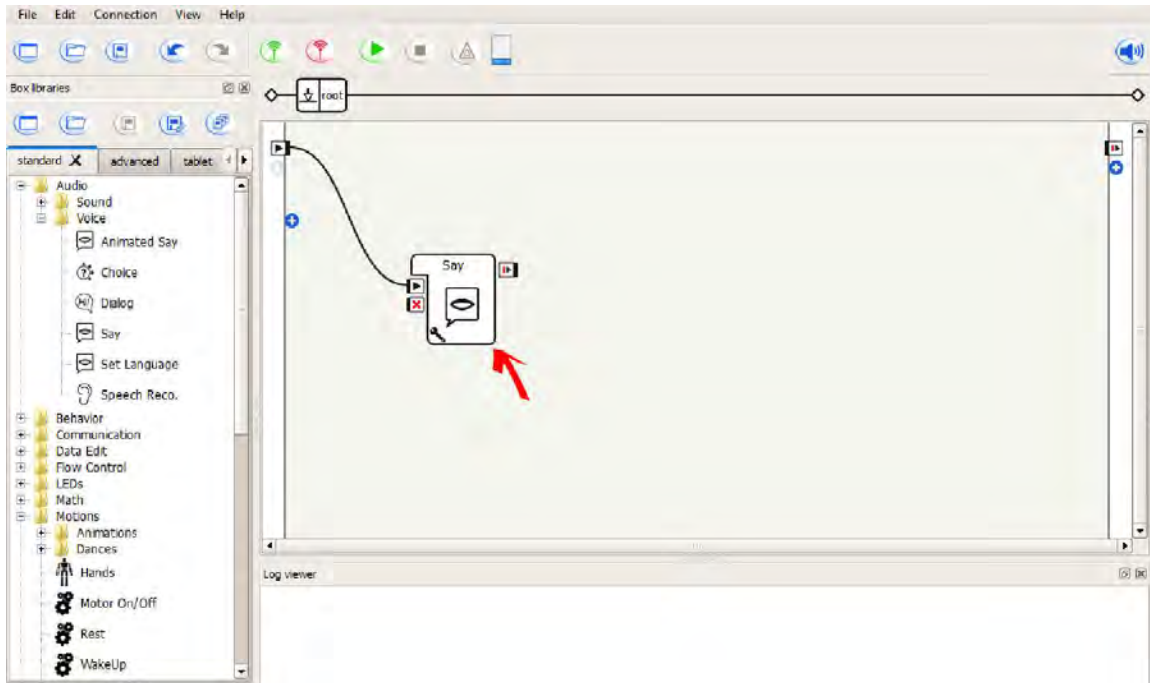
3. Finally, click the green “Play” button next to the “Connect” button. The NAO should say “Hello.”



Intermediate Task: Say Anything!

We will learn how to change the words the NAO says to “Hello world.”

1. Beginning with the results of the previous task, double-click on the “Say” box.



2. A “Localized Text”, will appear connected to a “Say Text” box. To return to the previous screen, click “root” on the upper box hierarchy “breadcrumb trail”.



3. Replace the “Hello” in the text box with “Hello World!”.

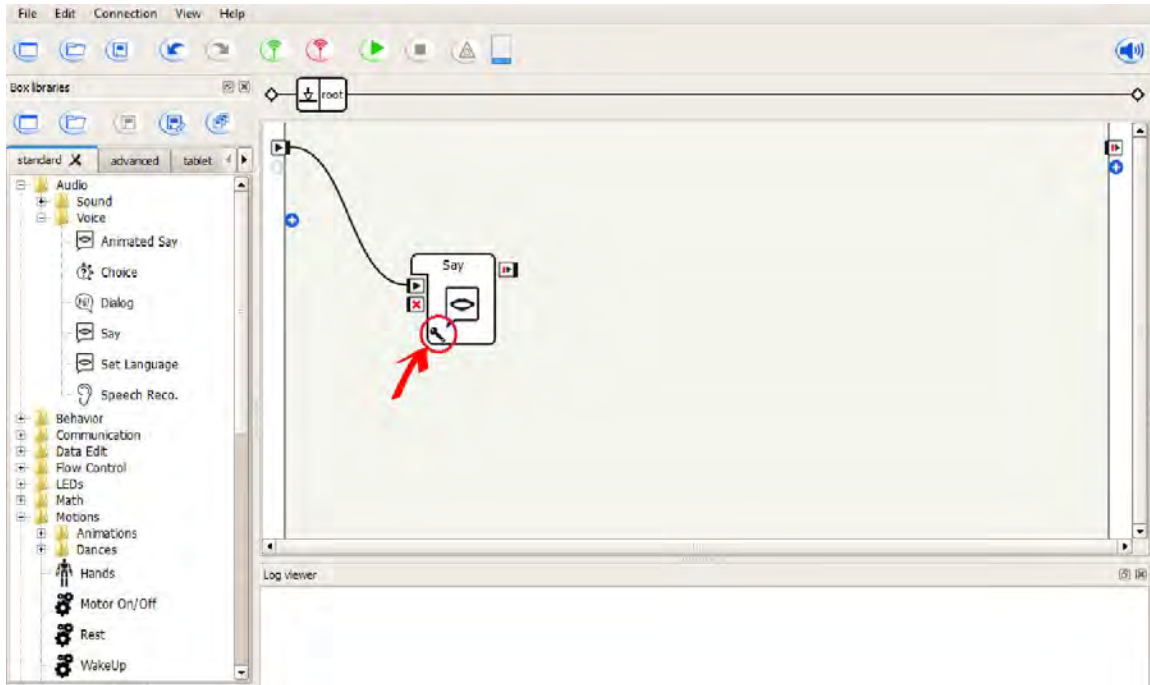


4. Click the play button, and listen to what the NAO says.
5. Try experimenting with different words and phrases.

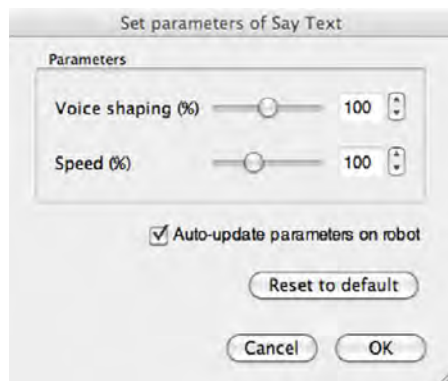
Intermediate Task: Voice Acting

The NAO can speak faster, slower, lower or higher, depending on two parameters, “Voice Shaping” and “Speed”.

1. Click on the wrench in the bottom left corner of the “Say” box.



2. The window below will appear:



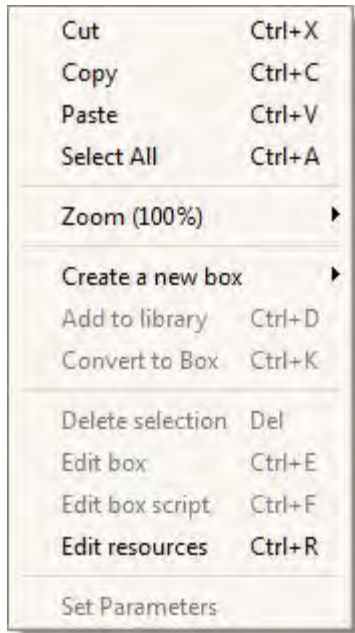
3. Move the two sliders to a different position and click OK. Press the play button for the NAO to speak again.
4. What changed? Try experimenting with different values for each of the sliders. What do the two parameters control?

Advanced Task: Speak with Python

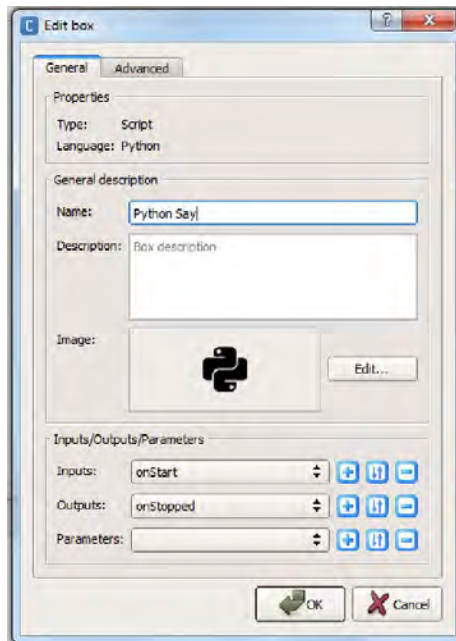
In addition to dragging and dropping boxes in Choregraphe, the NAO can be programmed using several programming languages including C++ and Python. We will be using Python for our exercises, but documentation about programming in C++ is available online.

We will begin with a simple programming exercise in Python, where we have the robot say “Hello World” exactly as we did before with the Choregraphe Say box. We will do this by creating our own box in Choregraphe that executes our python code.

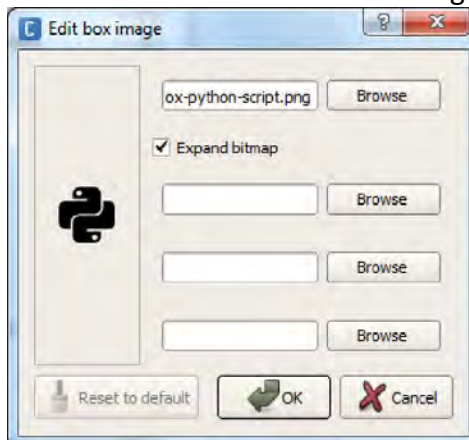
1. First, open an empty workspace in Choregraphe by creating a new project.
2. Right click on the workspace, and choose “Create a new box” from the menu.



3. A dialog will appear. In the “Name” box, enter “Python Say”, and in the tooltip box, enter a description of what your box will do. Do not change any of the other options.



4. Click the “Edit” button in the “Image” section. A new dialog box will appear.



5. Click the first “Browse” button and select the image “say.png”. This image will be displayed on the newly created box.

6. Click "OK" on both of the dialog windows. Your new box will appear in the workspace with the icon you selected.



7. Double click on the box to open the Script Editor. The Script Editor shows the Python code that is executed by your box.

```
class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self)

    def onLoad(self):
        #~ puts code for box initialization here
        pass

    def onUnload(self):
        #~ puts code for box cleanup here
        pass

    def onInput_onStart(self):
        #~ self.onStopped() #~ activate output of the box
        pass

    def onInput_onStop(self):
        self.onUnload() #~ it is recommended to call onUnload of this box in a onStop method, as the code
        #~ written in onUnload is used to stop the box as well
        pass
```

8. Look for the line that says “def onInput_onStart(self):”. This line is the definition for a *function* (or *method*) named onInput_onStart. A function is a procedure which can be called elsewhere in code. In this particular case, the onInput_onStart method is called when the box begins executing, and the indented code below this line is called. We will modify the code below this line to make the robot speak.
9. Replace pass in the onInput_onStart method with the following two lines of code, as in the picture below:
 ttsProxy = ALProxy(“ALTextToSpeech”)
 ttsProxy.say(“Hello world!”)

```
class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self)

    def onLoad(self):
        #~ puts code for box initialization here
        pass

    def onUnload(self):
        #~ puts code for box cleanup here
        pass

    def onInput_onStart(self):
        ttsProxy = ALProxy("ALTextToSpeech")
        ttsProxy.say("Hello world!")

    def onInput_onStop(self):
        self.onUnload() #~ it is recommended to call onUnload of this box in a onStop method, as the code
        #~ written in onUnload is used to stop the box as well
        pass
```

Note that in Python, the amount of *whitespace* (spaces and tabs) at the beginning of a line matters. Be sure to put the exact same combination of tabs and spaces on both of these lines (double the amount of whitespace of the line beginning with def).

The first line creates an object that gives us access to the robot’s text-to-speech capabilities. This object is assigned to the *variable* ttsProxy. We can access this object later through the name ttsProxy.

The second line calls a different function, say, belonging to the object we just created. This function takes an *argument*, “Hello world!”. “Hello world!” is a string, or a sequence of characters, denoted by the double quote marks. The robot will speak the string that was passed to the say method aloud.

10. Close the script editor window, and link your Python box to the start arrow.
11. Connect to the robot and press the play button. The robot should say “Hello world!” aloud.

Additional Exercises

1. Have the NAO introduce itself and greet the class.
2. Have the NAO play the parts of multiple characters in a play by using different voices for each character.
3. Have the NAO sing the alphabet by varying the voice shaping and pitch that it pronounces each letter with.

Module Questions

Basic:

1. How can you tell if the robot is charging?
2. How should you place the robot when turning it on?
3. How should the robot be held?
4. How is the order the Choregraphe boxes execute determined?

Intermediate:

5. What does the voice shaping parameter for the Say box control?
6. What does the speed parameter for the Say box control?

Advanced:

7. Name two programming languages that can be used on the NAO.
8. How can the NAO be programmed in Python using Choregraphe?
9. What is a *variable*?
10. What is a *function* or *method* in programming?
11. What is a function *argument*?
12. Do the number of spaces and tabs matter in python?