Getting Started with Python and Myro

Week #2
Prof. Ryan Kastner
Scribbler

- Three Wheels – Big ones on either side are powered by motors
- Scribbler’s movements are performed through the two motor-driven wheels

`motors(LEFT, RIGHT)`
`motors(0.0, 1.0) – only right motor works`
Which way does the robot move?
Speed of Myro

forward(SPEED)      forward(SPEED, SECONDS)
backward(SPEED)     backward(SPEED, SECONDS)
turnLeft(SPEED)     turnLeft(SPEED, SECONDS)
turnRight(SPEED)    turnRight(SPEED, SECONDS)
stop()              Make your robot traverse in a rectangular path

Try making the robot traverse along the shorter
two paths in the same time as the longer two
paths (sides) of the rectangle

T seconds

T seconds

T seconds
New Commands

- Making a robot carry out more complex behaviors requires several series of commands.
- Packaging a series of commands into a brand new command called a function.

Yoyo()

```python
def yoyo():
    forward(1)
    backward(1)
    stop()

>>> yoyo()
```

- Issuing the new function like this one in Python is called an **invocation**.
def yoyo():
    forward(1)
    wait(1)
    backward(1)
    wait(1)
    stop()

>>> def yoyo1(speed):
    forward(speed, 1)
    backward(speed, 1)

>>> yoyo1(0.5)

The Robot will move forward for 1 second before applying the command that follows the wait

The Robot moves at half the speed

Try passing other parameters like waitTime
Pass multiple parameters: yoyo3(0.5, 1.5)
Saving Commands in Modules

- When you work with different functions of the Robot you may end up with large collection of new functions.
- Wouldn’t it be better to store them in files on your computer so that you can call them when you need?
- Typical robot programs have huge number of functions and it is easy to store them in a file and call them when needed.
from myro import *
init()

# Define the new functions
def yoyo(speed, waitTime):
    forward(speed)
    wait(waitTime)
    backward(speed)
    wait(waitTime)
    stop()

- Click file on IDLE GUI and open a new window
- # is used to enter comments (so that you remember what you did when you come back to your program)
- Enter your program as you would do in your Python IDLE GUI
Writing and Saving to a File

- A good programmer always comments his program well
- Once you are done writing your program click File > Save As and enter the file name you like say ‘moves’ with an extension ‘.py’ (*All Python modules end with the filename extension .py*)
- Make sure they are always saved in the same folder as the Start Python.pyw file
You can use the file in two ways

```python
>>> from moves import *
# Try the commands we used before
>>> yoyo(0.5,0.5)
```

Accessing the commands defined in a module is similar to accessing the capabilities of the myro module

```python
from <MODULE NAME> import <SOMETHING>
When you replace <SOMETHING> with * you import everything
```
Functions as Building Blocks

- The basic syntax for defining a Python function takes the form:
  
  ```python
  def <FUNCTION NAME>(<PARAMETERS>):
    <SOMETHING>
    ...
    <SOMETHING>
  ```

- To define a new function,
  - Start by using the word `def` followed by the name of the function (`<FUNCTION NAME>`) followed by `<PARAMETERS>` enclosed in parenthesis followed by a colon (`:`).
  - This line is followed by the commands that make up the function definition (`<SOMETHING>...<SOMETHING>`).
Syntax

- Each command is to be placed on a separate line, and all lines that make up the definition should be indented (aligned) the same amount.
- The number of spaces in the indentation should be same.

```python
>>> def yoyo(speed, waitTime):
    forward(speed)
    wait(waitTime)
    backward(speed)
    wait(waitTime)
    stop()
SyntaxError: invalid syntax
```

Commands on the same line can be entered separated by a semi-colon (;)
Syntax – Readability in Python

- The indentation helps better readability of your code (program)
- Python also has a color highlighting feature
  *For example, the word def in a function definition appears in red, the name of your function, yoyo appears in blue*
- Defining new functions using existing functions is very effective and is used by all computer programmers
  *By defining the function yoyo as a new function using the existing functions (forward, backward, wait, stop)) you have abstracted a new behavior for your robot*
Sample Function using Functions

Try this:

```python
>>> def dance():
    yoyo(0.5, 0.5)
    yoyo(0.5, 0.5)
    wiggle(0.5, 1)
    wiggle(0.5, 1)

>>> dance()
```
Summary

- Commands to make a Robot move in different ways
- Define new commands by defining new Python Functions
- Saving Functions in a File
- Importing from a File or Module
Try These

Compare the robot's movements in the commands turnLeft(1), turnRight(1) and rotate(1), rotate(-1).

Closely observe the robot's behavior and then also try the motor commands:

```python
>>> motors(-0.5, 0.5)
>>> motors(0.5, -0.5)
>>> motors(0, 0.5)
>>> motors(0.5, 0)
```