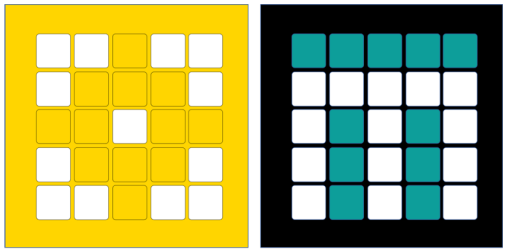
PRIME LESSONS

By the Makers of EV3Lessons



IMPORTING CUSTOM LIBRARIES

BY SANJAY AND ARVIND SESHAN

This lesson uses SPIKE 3 software

LESSON OBJECTIVES (ADVANCED)

Learn how to input your own libraries of functions in your MicroPython programs

WARNING:

The sample code is provided as-is and without any warranties.

This lesson involves reading and writing files from the hub's flash drive. Accidentally deleting files may put your hub in a state where the OS has to be reinstalled and may cause loss of data on the hub.

Use at your own risk.

Copyright © 2021 Prime Lessons (primelessons.org) CC-BY-NC-SA. (Last edit: 09/13/2023)

WHY IS THIS USEFUL

You do not have to copy and paste functions that you create into every program that you write. You can just import your functions into each program.

When you make a change to your functions, you do not need to change it in every program. You can just change it in the library and re-export the library.

Copyright © 2021 Prime Lessons (primelessons.org) CC-BY-NC-SA. (Last edit: 09/13/2023)

STEP 1: CREATING THE FUNCTION LIBRARY

Create a new project file

This project will contain your library of functions.

In this example, we will create two simple functions. One that shows a left arrow on the light matrix, and one that shows a left arrow

The main function will test these library functions

Download the project to a slot on the hub that you don't use for missions. We used slot 19.

Copyright © 2021 Prime Lessons (primelessons.org) CC-BY-NC-SA. (Last edit: 09/13/2023)

STEP 1: FUNCTION LIBRARY CODE

```
# Start library code
from hub import light_matrix

def right():
    light_matrix.show_image(light_matrix.IMAGE_ARROW_E)

def left():
    light_matrix.show_image(light_matrix.IMAGE_ARROW_W)
# End Library code

# Start test code
async def main():
    right()
    await runloop.sleep_ms(1000)
    left()

import runloop, sys
runloop.run(main())
sys.exit("Done test")
#End test code
```

Copyright © 2021 Prime Lessons (primelessons.org) CC-BY-NC-SA. (Last edit: 09/13/2023)

STEP 2: EXPORTING THE LIBRARY

In SP3, you can import a text-based Python file (.py) that is present in the hub root.

We will write a second program that exports the first program's library functions to a file called "customlib.py" in the SP root folder ("/flash").

Use a slot that you don't use for missions. We used slot 18.

It will first create a string that contains all the library functions. Test code is excluded.

It then writes the string out to a file in the SPIKE hub root.

It does not itself import or use the library. It is simply an exporter utility.

TIP: You can also create multiple libraries by exporting chunks of code to different .py files in the root

Copyright © 2021 Prime Lessons (primelessons.org) CC-BY-NC-SA. (Last edit: 09/13/2023)

STEP 2: EXPORT CODE

```
# Library code string using python multiline quotes. Do not include test code, only the functions you want to reuse and the imports they need.
code: str = '''
from hub import light_matrix

def right():
    light_matrix.show_image(light_matrix.IMAGE_ARROW_E)

def left():
    light_matrix.show_image(light_matrix.IMAGE_ARROW_W)
'''

def exportProgram(): # Function to export the library code string
    import os
    global code
    os.chdir('/flash') # change directory to root
    try:
        os.remove('customlib.py') # remove any existing library file of the same name
    except:
        pass
    f = open('customlib.py', 'w+') # Create a new file customlib.py in the SPIKE hub root
    f.write(code) # Write out the library code string to the customlib.py file
    f.close()

import sys
exportProgram() # Runs the export function
sys.exit("Export complete")
```

Copyright © 2021 Prime Lessons (primelessons.org) CC-BY-NC-SA. (Last edit: 09/13/2023)

STEP 3: USING THE LIBRARY

You can now import the library in any of your other programs

Copyright © 2021 Prime Lessons (primelessons.org) CC-BY-NC-SA. (Last edit: 09/13/2023)

STEP 3: IMPORT CODE EXAMPLE

```
import runloop, sys
# customlib can be imported easily just like any other import.
import customlib

async def main():
    customlib.left() # use customlib functions by prefixing with customlib.
    await runloop.sleep_ms(1000)
    customlib.right()
    await runloop.sleep_ms(1000)
    sys.exit('Done') # customlib

runloop.run(main())

# Alternately, if you don't want to use prefixes, you can do:
from customlib import *
left()
right()
```

Copyright © 2021 Prime Lessons (primelessons.org) CC-BY-NC-SA. (Last edit: 09/13/2023)

CREDITS

This lesson was created by Sanjay Seshan and Arvind Seshan for Prime Lessons

Additional contributions by FLL Share & Learn community members

SP3 specific tips provided by Ethan Danahy

More lessons are available at www.primelessons.org

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

Copyright © 2021 Prime Lessons (primelessons.org) CC-BY-NC-SA. (Last edit: 09/13/2023)