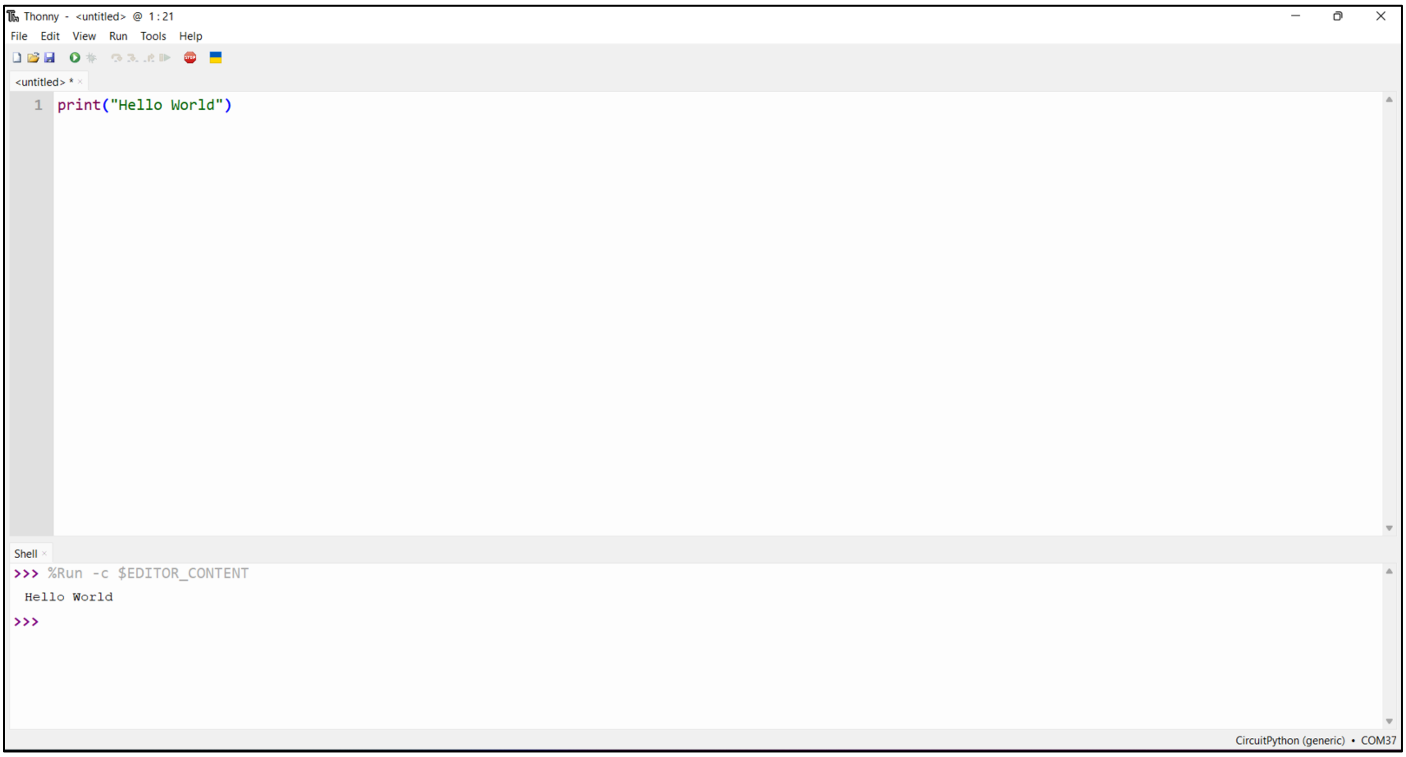
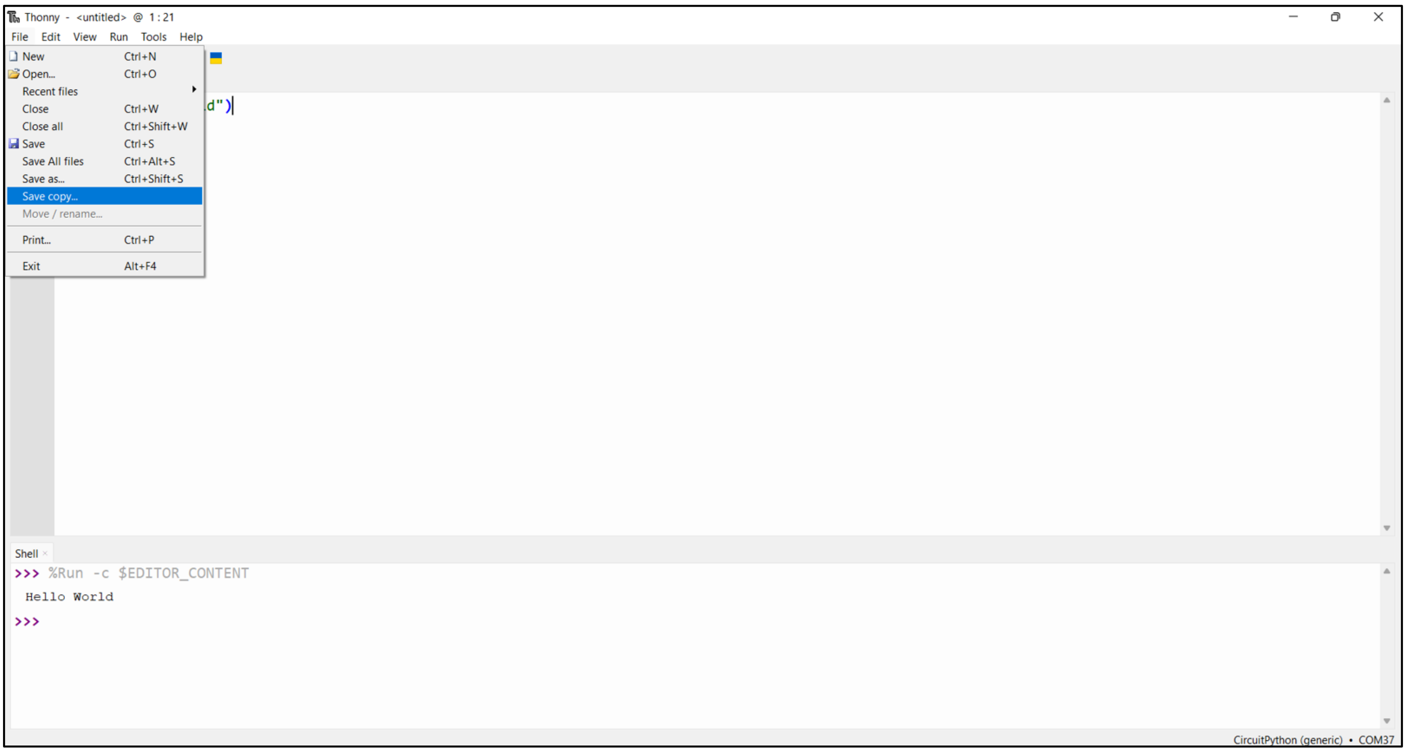
Steps: To move script file inside Raspberry Pi Pico W of NetPi



```
print("Hello World")
```

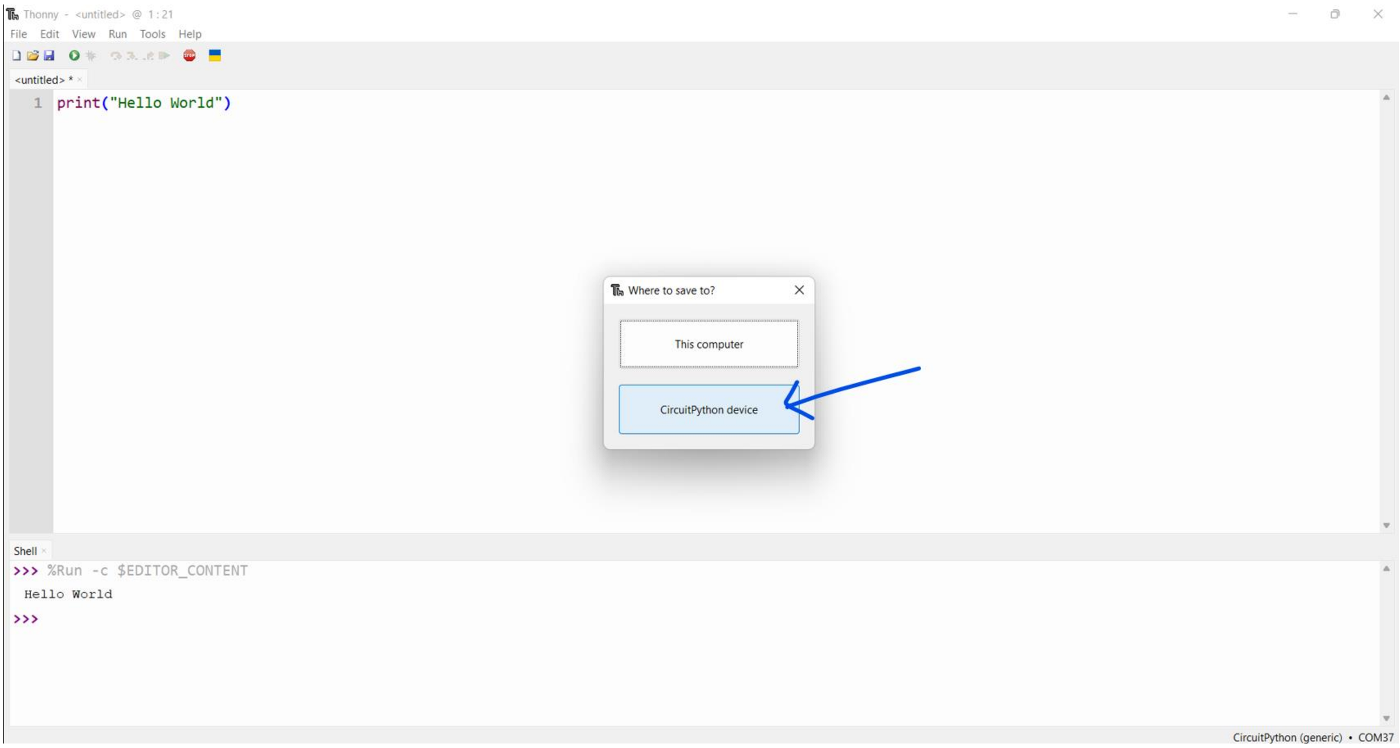
```
>>> %Run -c $EDITOR_CONTENT
 Hello World
>>>
```

- Above example code is opened in thonny IDE and run with NetPi’s Pico but to move this file on Pico click on File -> Save copy.. and Select the CircuitPython device



```
d")
```

```
>>> %Run -c $EDITOR_CONTENT
 Hello World
>>>
```

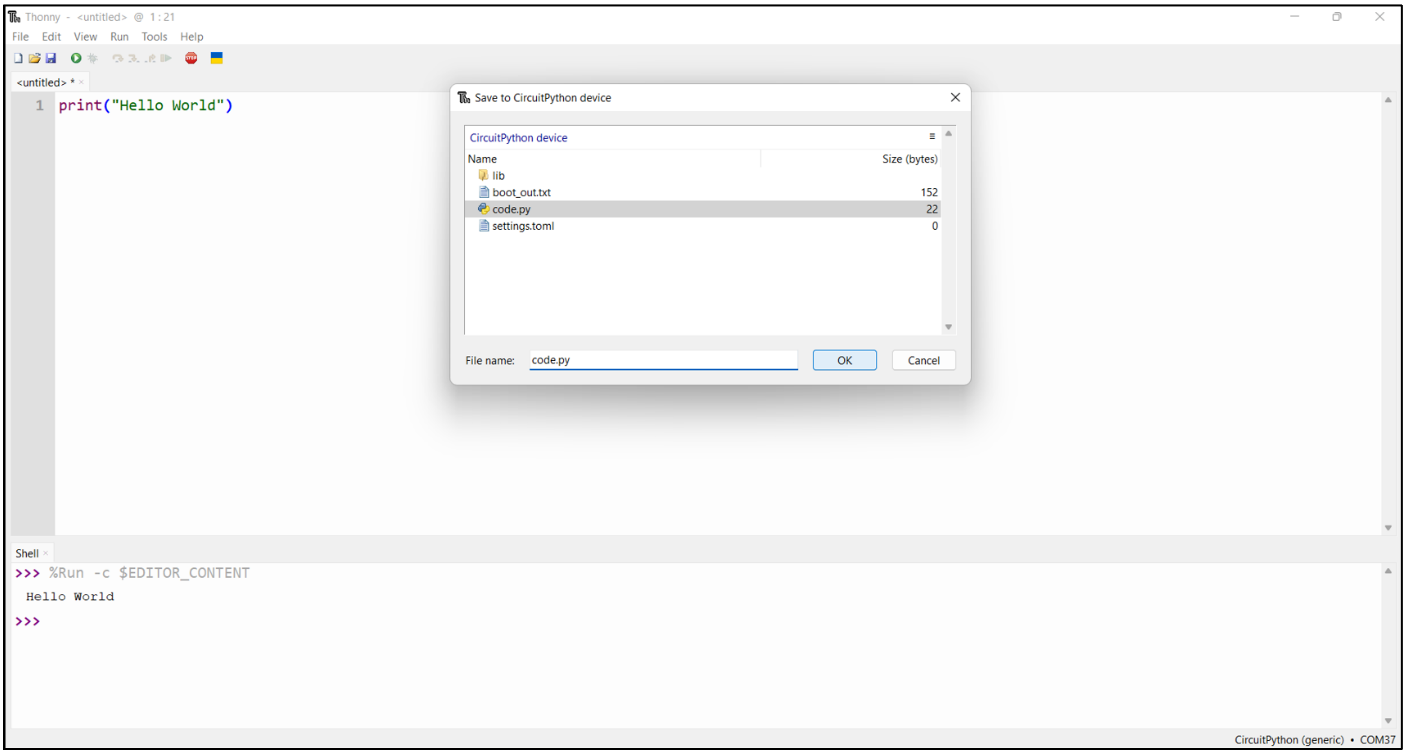


Thonny - <untitled> @ 1 : 21
File Edit View Run Tools Help
<untitled> *
Where to save to?
This computer
CircuitPython device
Shell
CircuitPython (generic) • COM37

```
print("Hello World")
```

```
>>> %Run -c $EDITOR_CONTENT
 Hello World
>>>
```

- Save file with name "code.py". Already you will find default file there with same name then overwrite it.



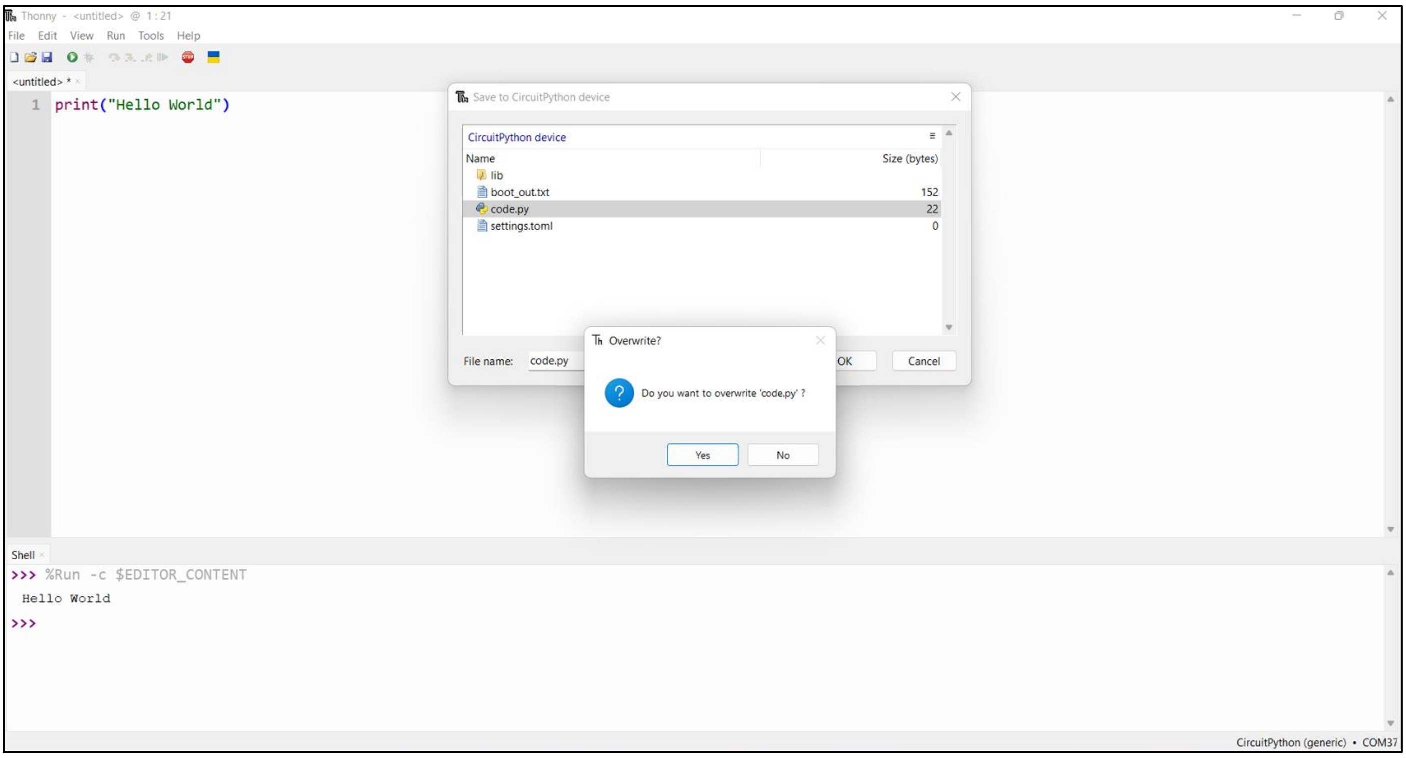
Thonny - <untitled> @ 1 : 21
File Edit View Run Tools Help
<untitled> *
Save to CircuitPython device
CircuitPython device
OK
Cancel
Shell
CircuitPython (generic) • COM37

```
print("Hello World")
```

Name	Size (bytes)
lib	
boot_out.txt	152
code.py	22
settings.toml	0

File name: code.py

```
>>> %Run -c $EDITOR_CONTENT
 Hello World
>>>
```


Thonny - <untitled> @ 1 : 21
File Edit View Run Tools Help
<untitled> *
Save to CircuitPython device
CircuitPython device
Name
Size (bytes)
File name:
OK
Cancel
Overwrite?
Yes
No
Shell
CircuitPython (generic) • COM37

```
print("Hello World")
```

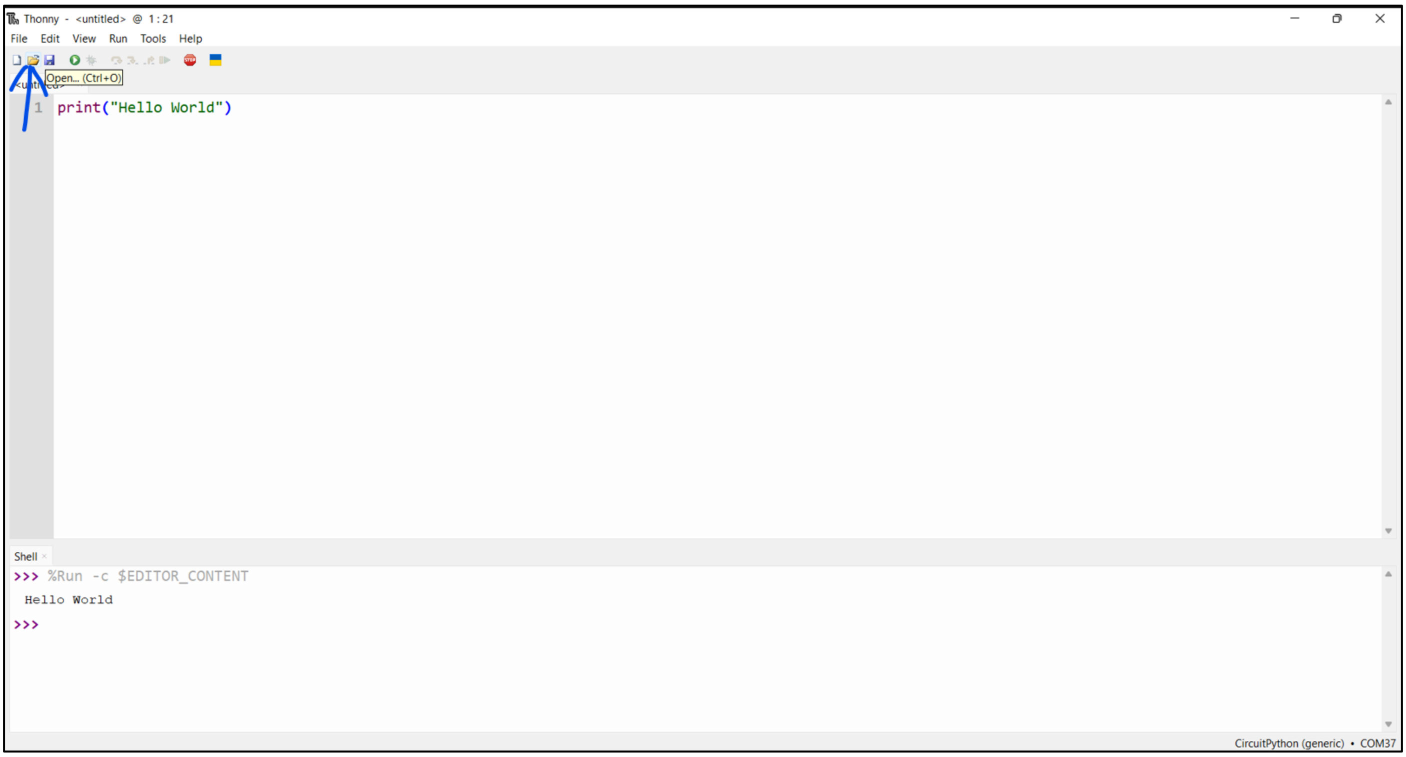
Name	Size (bytes)
lib	
boot_out.txt	152
code.py	22
settings.toml	0

File name: code.py

Do you want to overwrite 'code.py' ?

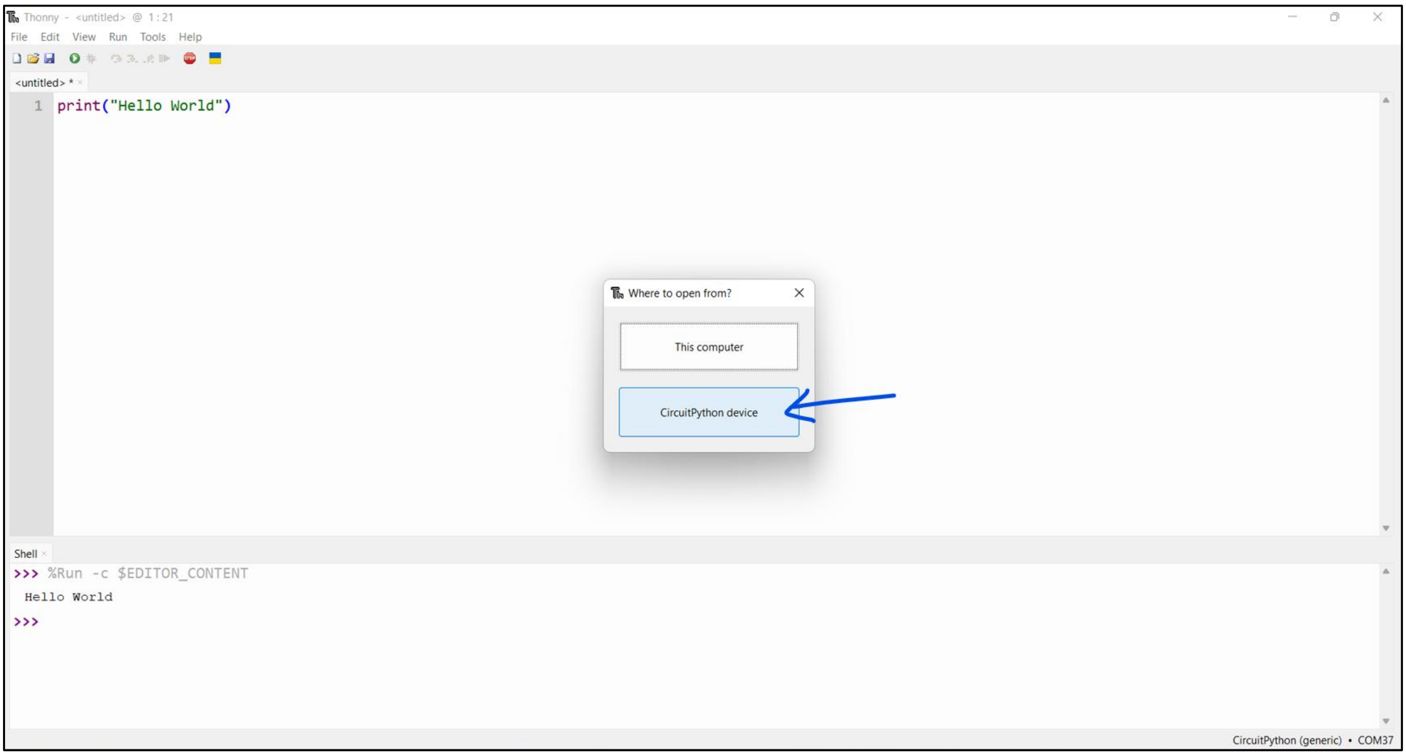
```
>>> %Run -c $EDITOR_CONTENT
 Hello World
>>>
```

- You can verify if file is moved or not. For this just click open folder icon, and again select CircuitPython device


Thonny - <untitled> @ 1 : 21
File Edit View Run Tools Help
Open... (Ctrl+O)
Shell
CircuitPython (generic) • COM37

```
print("Hello World")
```

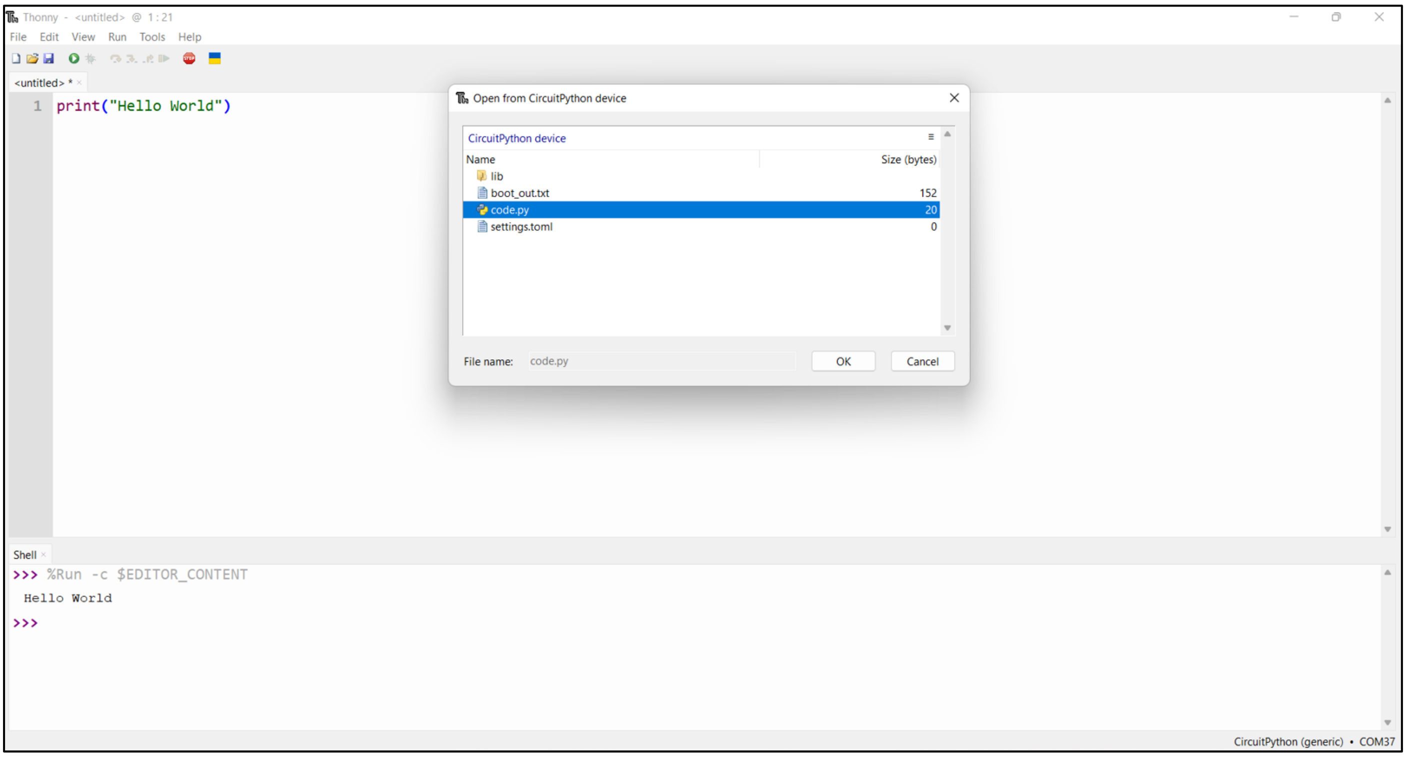
```
>>> %Run -c $EDITOR_CONTENT
 Hello World
>>>
```


Thonny - <untitled> @ 1 : 21
File Edit View Run Tools Help
<untitled> *
Where to open from?
This computer
CircuitPython device
Shell
CircuitPython (generic) • COM37

```
print("Hello World")
```

```
>>> %Run -c $EDITOR_CONTENT
  Hello World
>>>
```

- Open code.py file from device and see [code.py] bracket which indicates file from inside Pico.

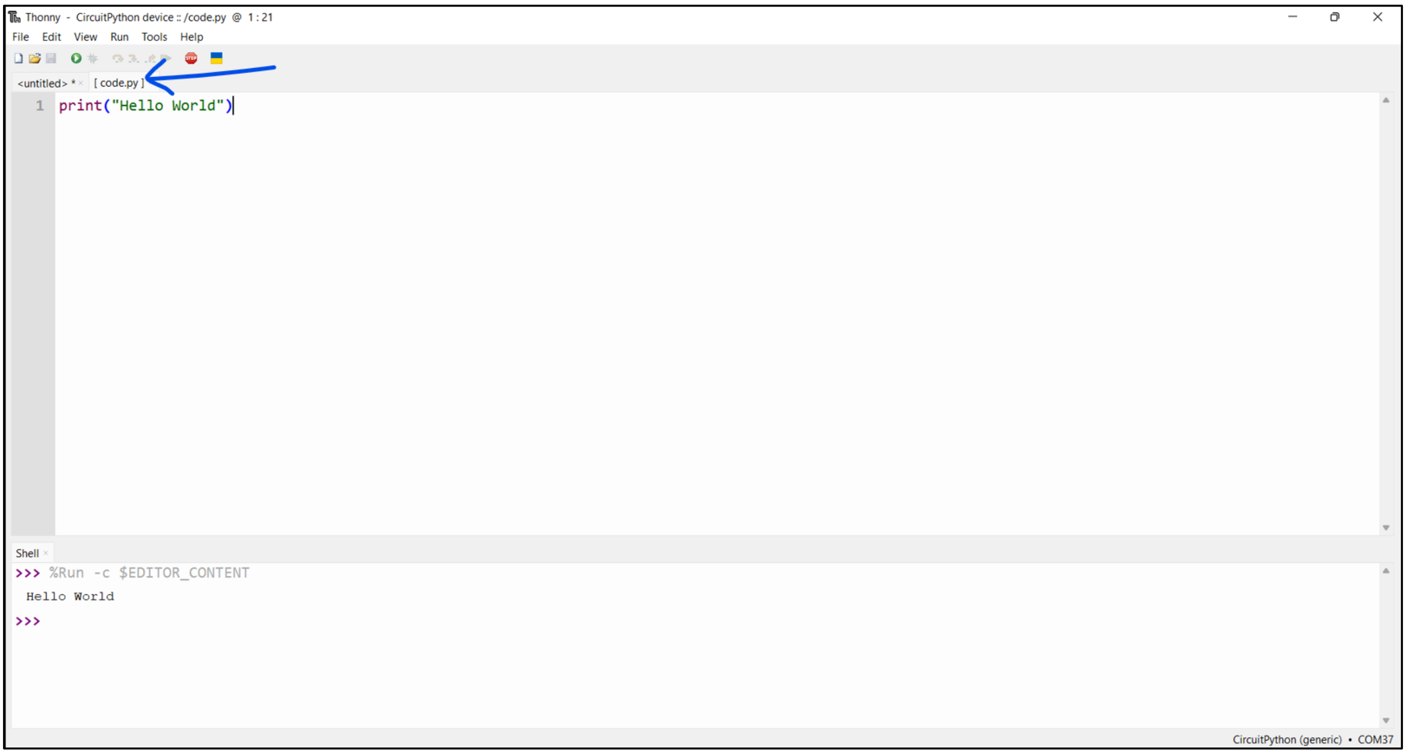

Thonny - <untitled> @ 1 : 21
File Edit View Run Tools Help
<untitled> *
Open from CircuitPython device
CircuitPython device
OK
Cancel
Shell
CircuitPython (generic) • COM37

Name	Size (bytes)
lib	
boot_out.txt	152
code.py	20
settings.toml	0

File name: code.py

```
print("Hello World")
```

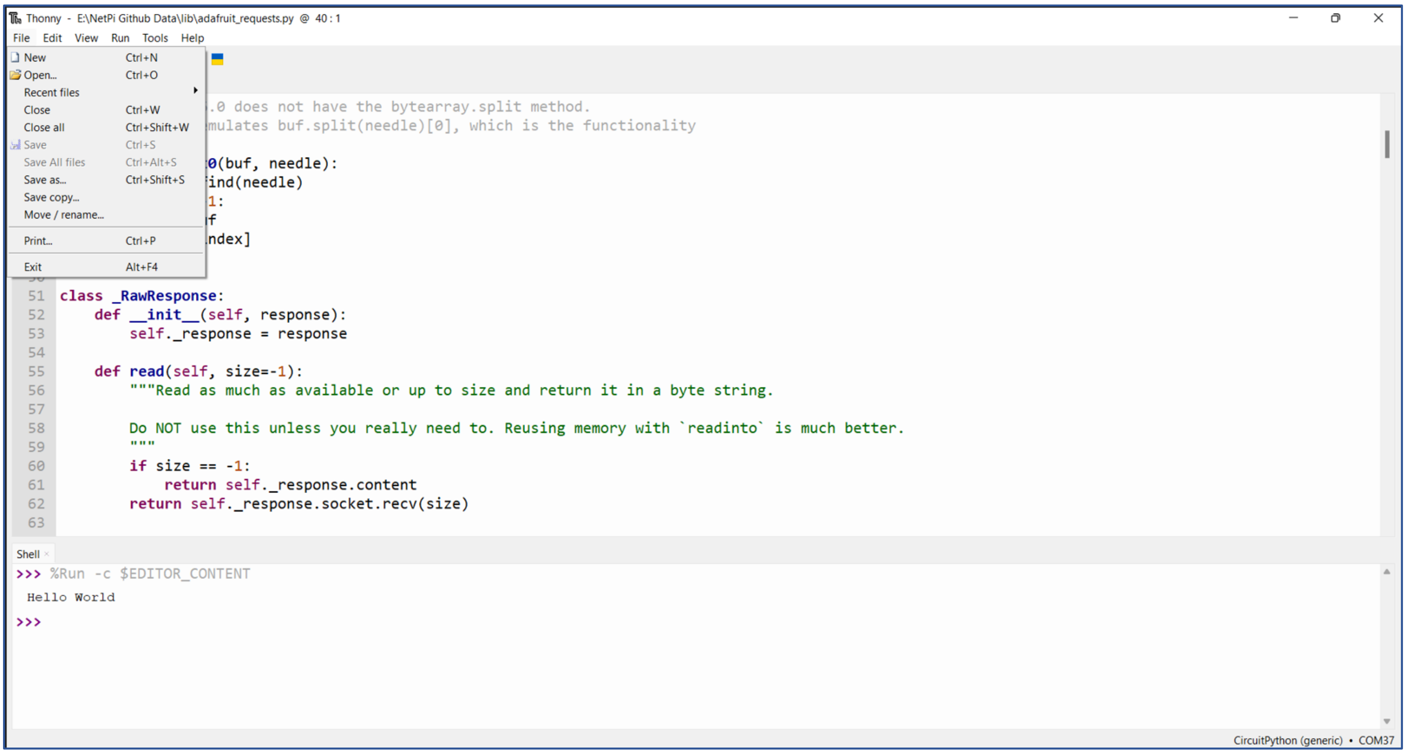
```
>>> %Run -c $EDITOR_CONTENT
  Hello World
>>>
```


Thonny - CircuitPython device :: /code.py @ 1 : 21
File Edit View Run Tools Help
<untitled> *
[code.py]
Shell
CircuitPython (generic) • COM37

```
print("Hello World")
```

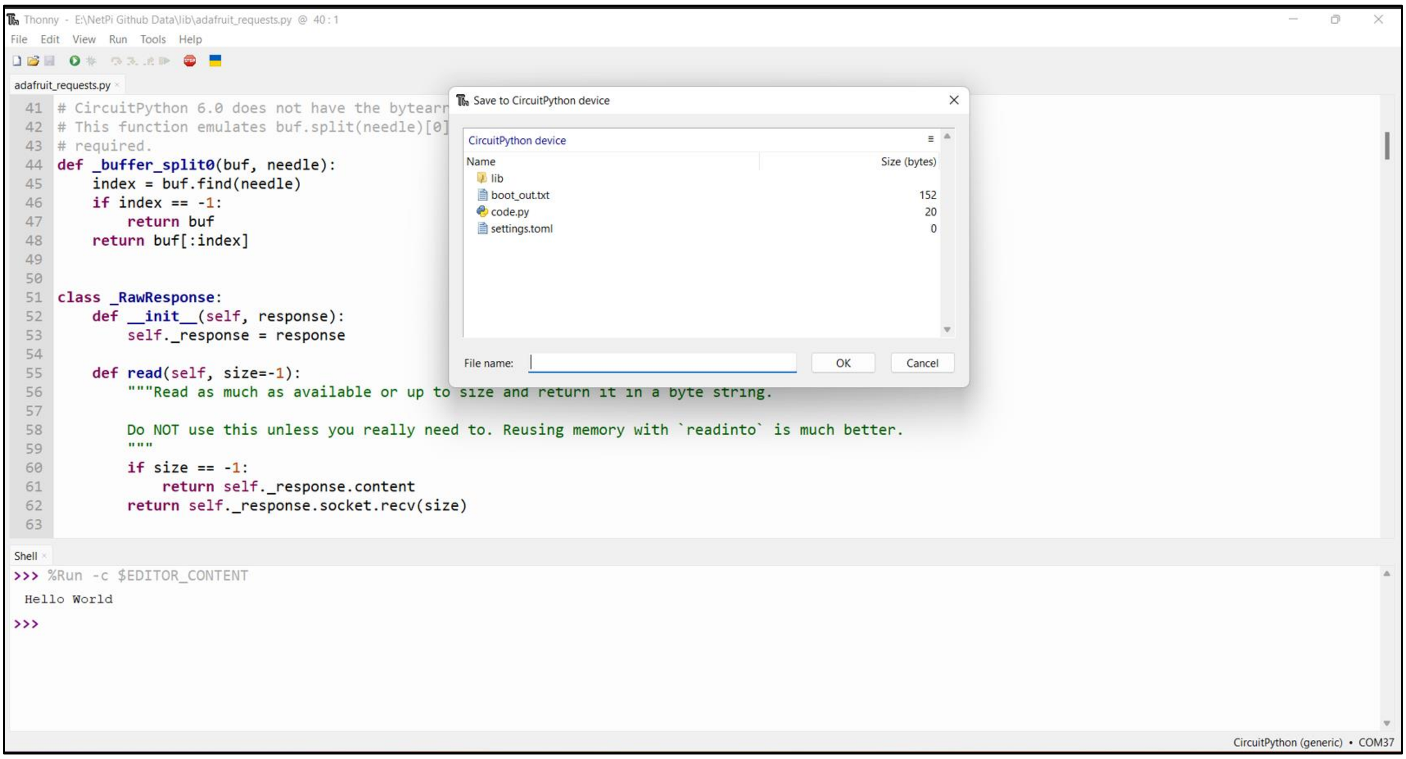
```
>>> %Run -c $EDITOR_CONTENT
 Hello World
>>>
```

Checkout below steps to copy files inside **lib** folder **of device**, process is same ->



```
.0 does not have the bytearray.split method.
mulates buf.split(needle)[0], which is the functionality

0(buf, needle):
ind(needle)
1:
f
ndex]

class _RawResponse:
    def __init__(self, response):
        self._response = response

    def read(self, size=-1):
        """Read as much as available or up to size and return it in a byte string.

        Do NOT use this unless you really need to. Reusing memory with `readinto` is much better.
        """
        if size == -1:
            return self._response.content
        return self._response.socket.recv(size)

```

```
>>> %Run -c $EDITOR_CONTENT
 Hello World
>>>
```



Thonny - E:\NetPi Github Data\lib\adafruit_requests.py @ 40 : 1
File Edit View Run Tools Help
adafruit_requests.py
Shell
CircuitPython (generic) • COM37

```
# CircuitPython 6.0 does not have the bytearr
# This function emulates buf.split(needle)[0]
# required.
def _buffer_split0(buf, needle):
    index = buf.find(needle)
    if index == -1:
        return buf
    return buf[:index]


class _RawResponse:
    def __init__(self, response):
        self._response = response

    def read(self, size=-1):
        """Read as much as available or up to size and return it in a byte string.

        Do NOT use this unless you really need to. Reusing memory with `readinto` is much better.
        """
        if size == -1:
            return self._response.content
        return self._response.socket.recv(size)

```

Save to CircuitPython device

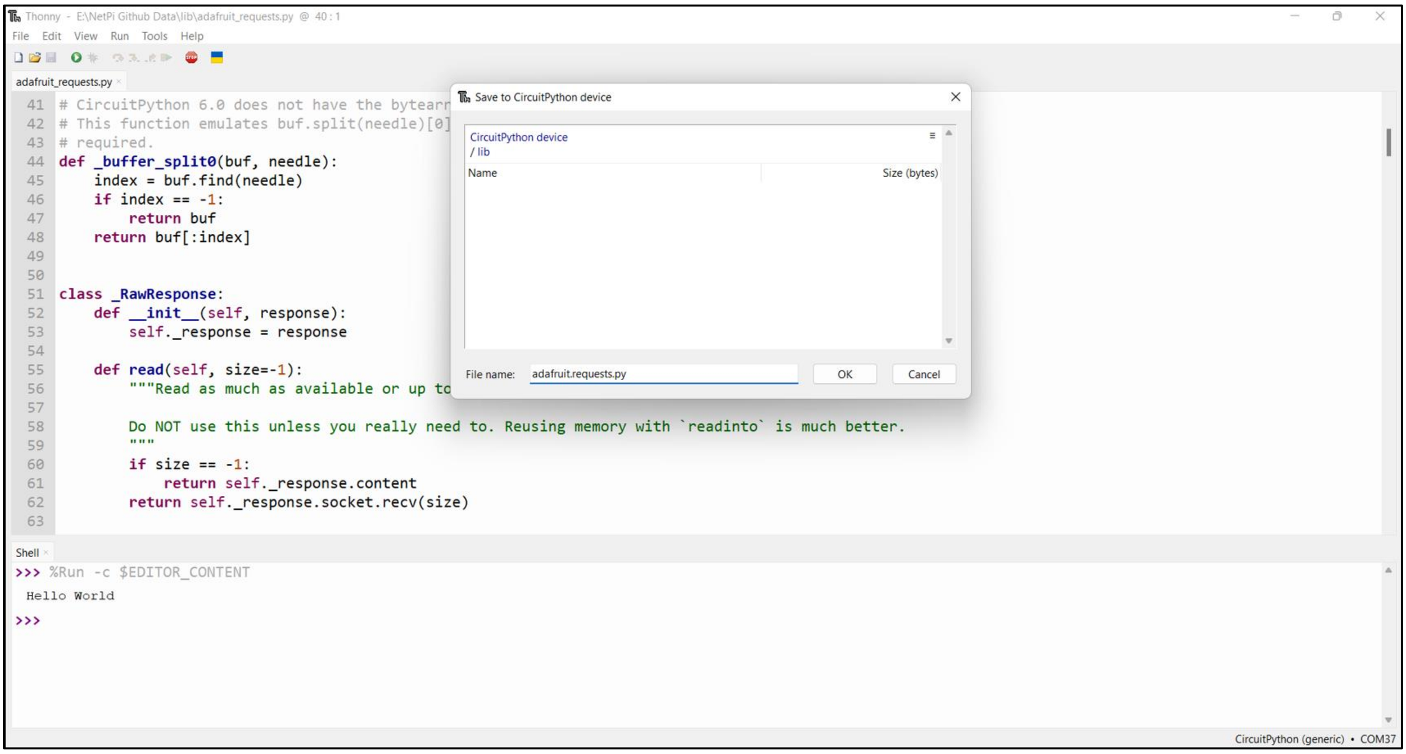
CircuitPython device

```
Name                                   Size (bytes)
lib
boot_out.txt                                    152
code.py                                          20
settings.toml                                     0
```

File name:

OK Cancel

```
>>> %Run -c $EDITOR_CONTENT
 Hello World
>>>
```



Thonny - E:\NetPi Github Data\lib\adafruit_requests.py @ 40 : 1
File Edit View Run Tools Help
adafruit_requests.py
Shell
CircuitPython (generic) • COM37

```
# CircuitPython 6.0 does not have the bytearr
# This function emulates buf.split(needle)[0]
# required.
def _buffer_split0(buf, needle):
    index = buf.find(needle)
    if index == -1:
        return buf
    return buf[:index]


class _RawResponse:
    def __init__(self, response):
        self._response = response

    def read(self, size=-1):
        """Read as much as available or up to

        Do NOT use this unless you really need to. Reusing memory with `readinto` is much better.
        """
        if size == -1:
            return self._response.content
        return self._response.socket.recv(size)

```

Save to CircuitPython device

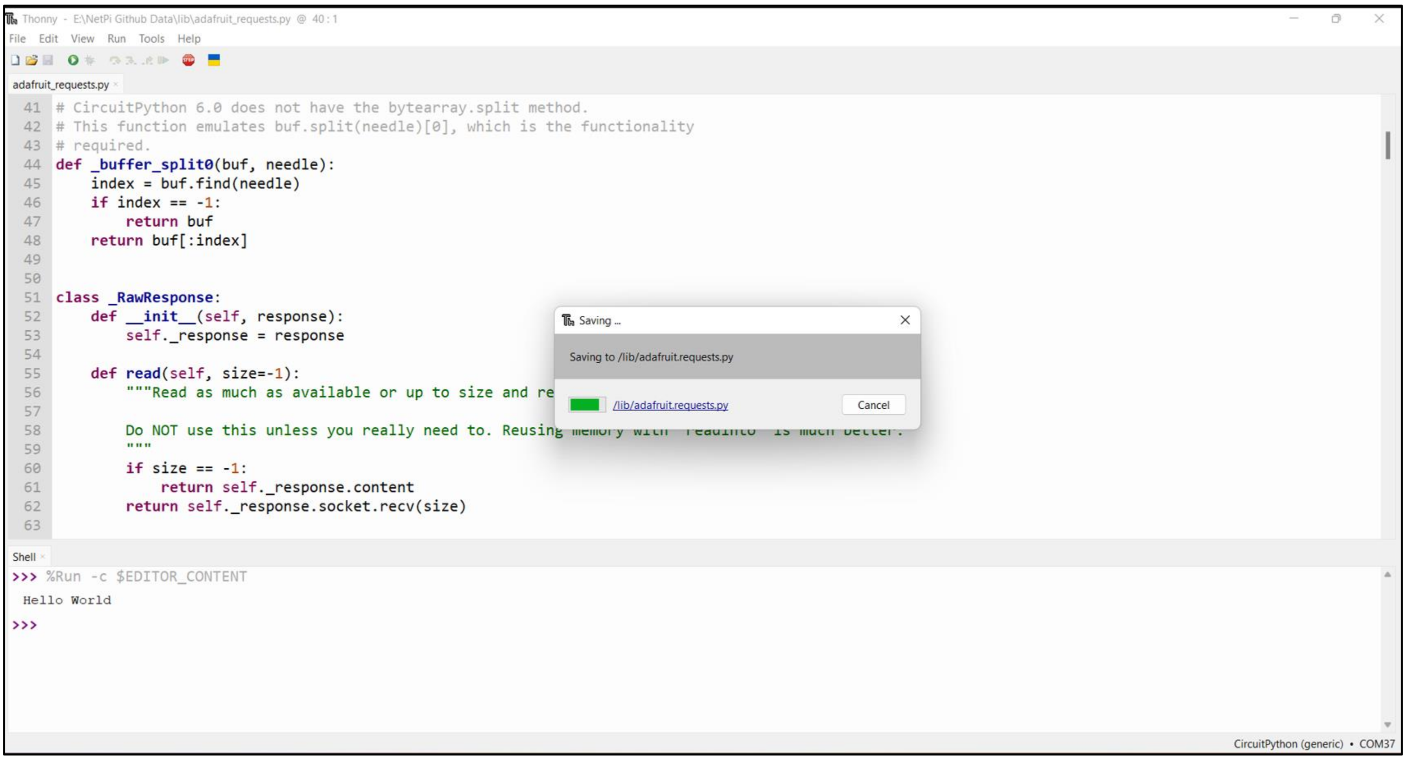
CircuitPython device
/ lib

```
Name                                   Size (bytes)
```

File name: adafruit.requests.py

OK Cancel

```
>>> %Run -c $EDITOR_CONTENT
 Hello World
>>>
```


Thonny - E:\NetPi Github Data\lib\adafruit_requests.py @ 40 : 1
File Edit View Run Tools Help
adafruit_requests.py
Shell

```
# CircuitPython 6.0 does not have the bytearray.split method.
# This function emulates buf.split(needle)[0], which is the functionality
# required.
def _buffer_split0(buf, needle):
    index = buf.find(needle)
    if index == -1:
        return buf
    return buf[:index]


class _RawResponse:
    def __init__(self, response):
        self._response = response

    def read(self, size=-1):
        """Read as much as available or up to size and re

        Do NOT use this unless you really need to. Reusing memory with `readinto` is much better.
        """
        if size == -1:
            return self._response.content
        return self._response.socket.recv(size)
```

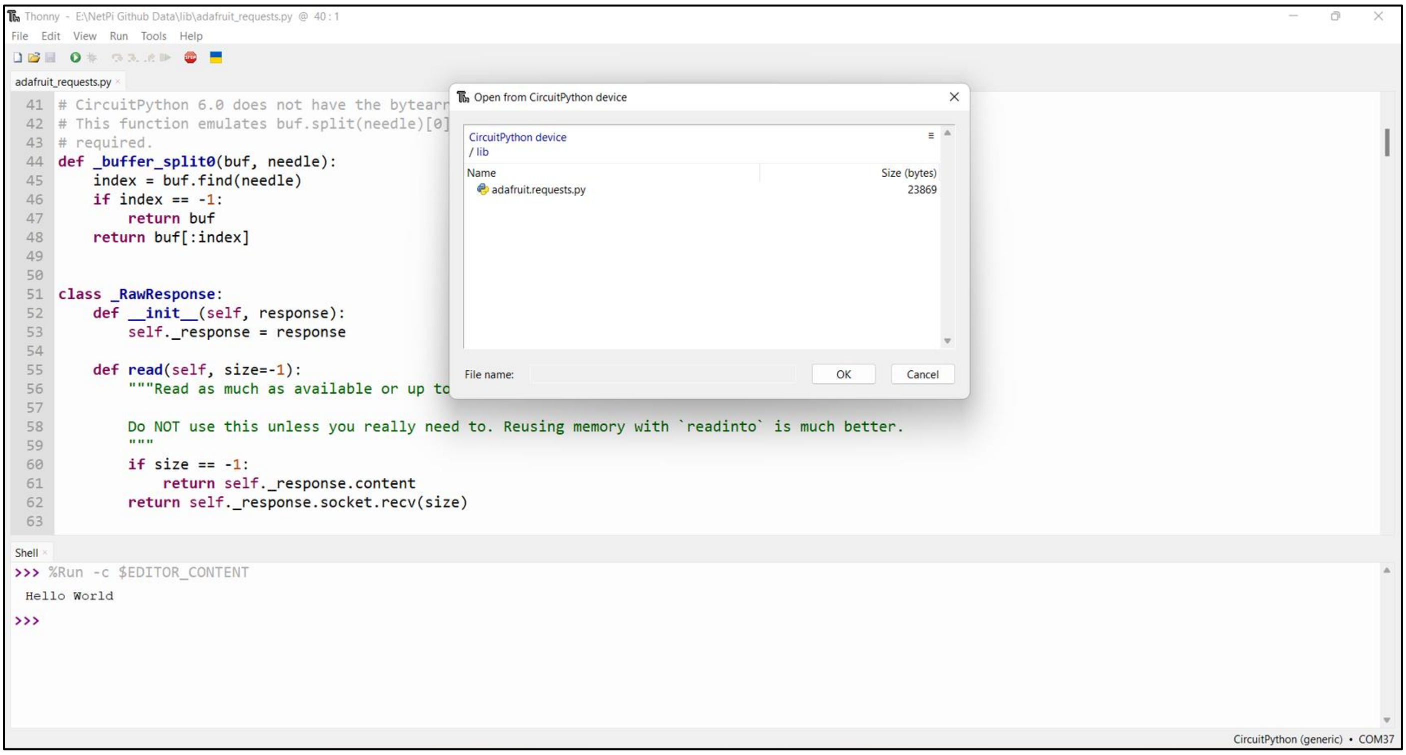
Saving ...

Saving to /lib/adafruit.requests.py

/lib/adafruit.requests.py

```
>>> %Run -c $EDITOR_CONTENT
 Hello World
>>>
```

CircuitPython (generic) • COM37


Thonny - E:\NetPi Github Data\lib\adafruit_requests.py @ 40 : 1
File Edit View Run Tools Help
adafruit_requests.py
Shell

```
# CircuitPython 6.0 does not have the bytearr
# This function emulates buf.split(needle)[0]
# required.
def _buffer_split0(buf, needle):
    index = buf.find(needle)
    if index == -1:
        return buf
    return buf[:index]


class _RawResponse:
    def __init__(self, response):
        self._response = response

    def read(self, size=-1):
        """Read as much as available or up to

        Do NOT use this unless you really need to. Reusing memory with `readinto` is much better.
        """
        if size == -1:
            return self._response.content
        return self._response.socket.recv(size)
```

Open from CircuitPython device

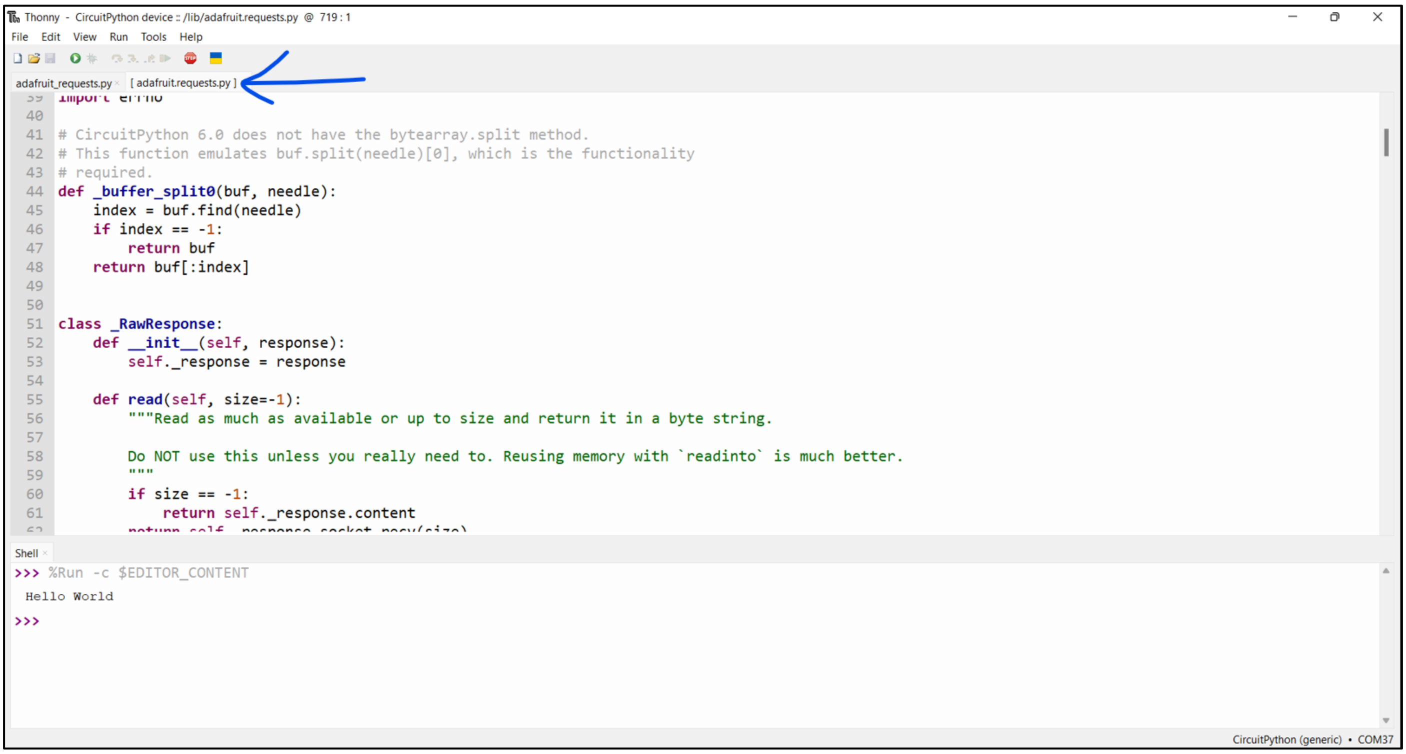
CircuitPython device
/ lib

Name	Size (bytes)
adafruit.requests.py	23869

File name:

```
>>> %Run -c $EDITOR_CONTENT
 Hello World
>>>
```

CircuitPython (generic) • COM37


Thonny - CircuitPython device :: /lib/adafruit.requests.py @ 719 : 1
File Edit View Run Tools Help
adafruit_requests.py
[adafruit.requests.py]
Shell
CircuitPython (generic) • COM37

```
import errno

# CircuitPython 6.0 does not have the bytearray.split method.
# This function emulates buf.split(needle)[0], which is the functionality
# required.
def _buffer_split0(buf, needle):
    index = buf.find(needle)
    if index == -1:
        return buf
    return buf[:index]


class _RawResponse:
    def __init__(self, response):
        self._response = response

    def read(self, size=-1):
        """Read as much as available or up to size and return it in a byte string.

        Do NOT use this unless you really need to. Reusing memory with `readinto` is much better.
        """
        if size == -1:
            return self._response.content
        return self._response.socket.recv(size)
```

```
>>> %Run -c $EDITOR_CONTENT
 Hello World
>>>
```

[] around file confirms that file successfully transferred to device.

Similarly, copy all script files of lib(https://github.com/sbcshop/NetPi_Pico_Ethernet/tree/main/lib) folder from github to device lib folder before try any of code otherwise file error will pops up.