

\*\*\*\*\*Default\*\*\*\*\*

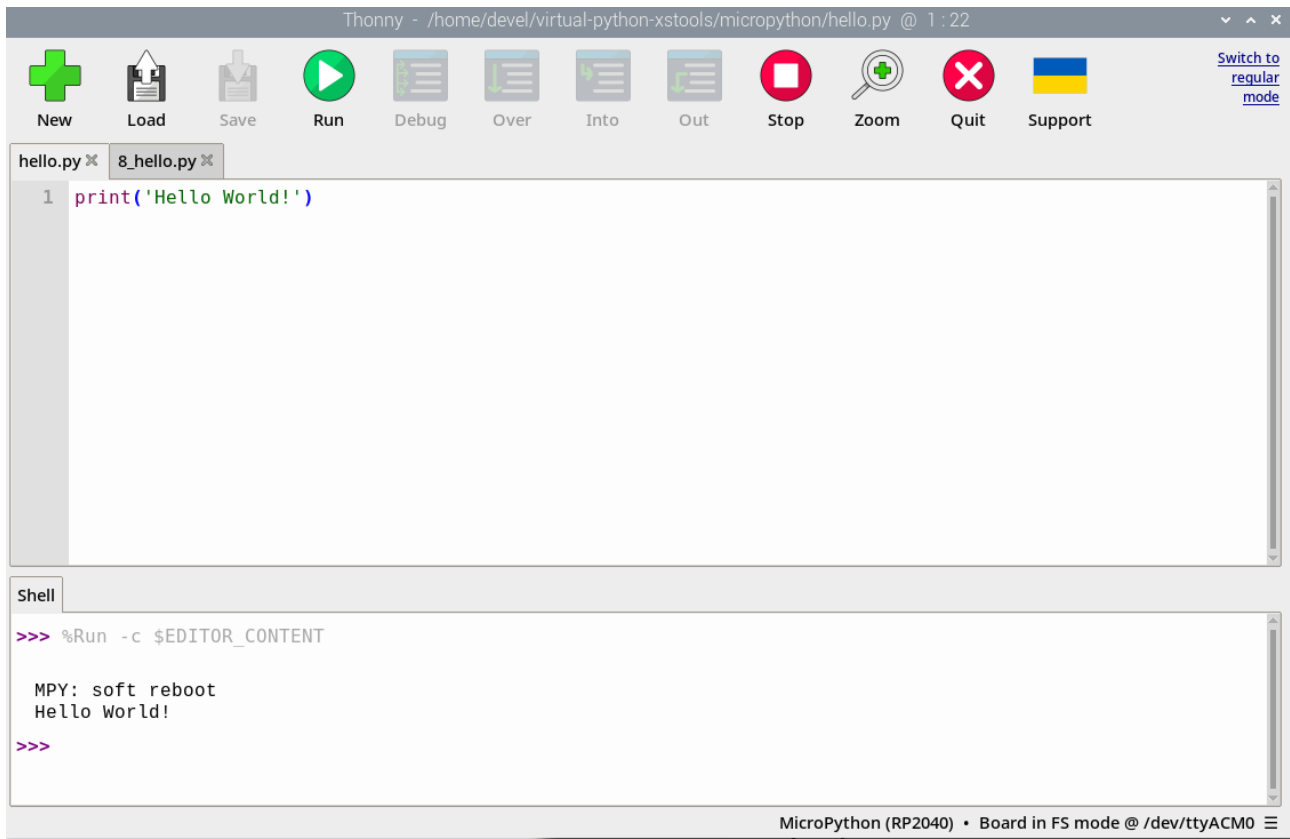
## MicroPython Pico

11/09/24

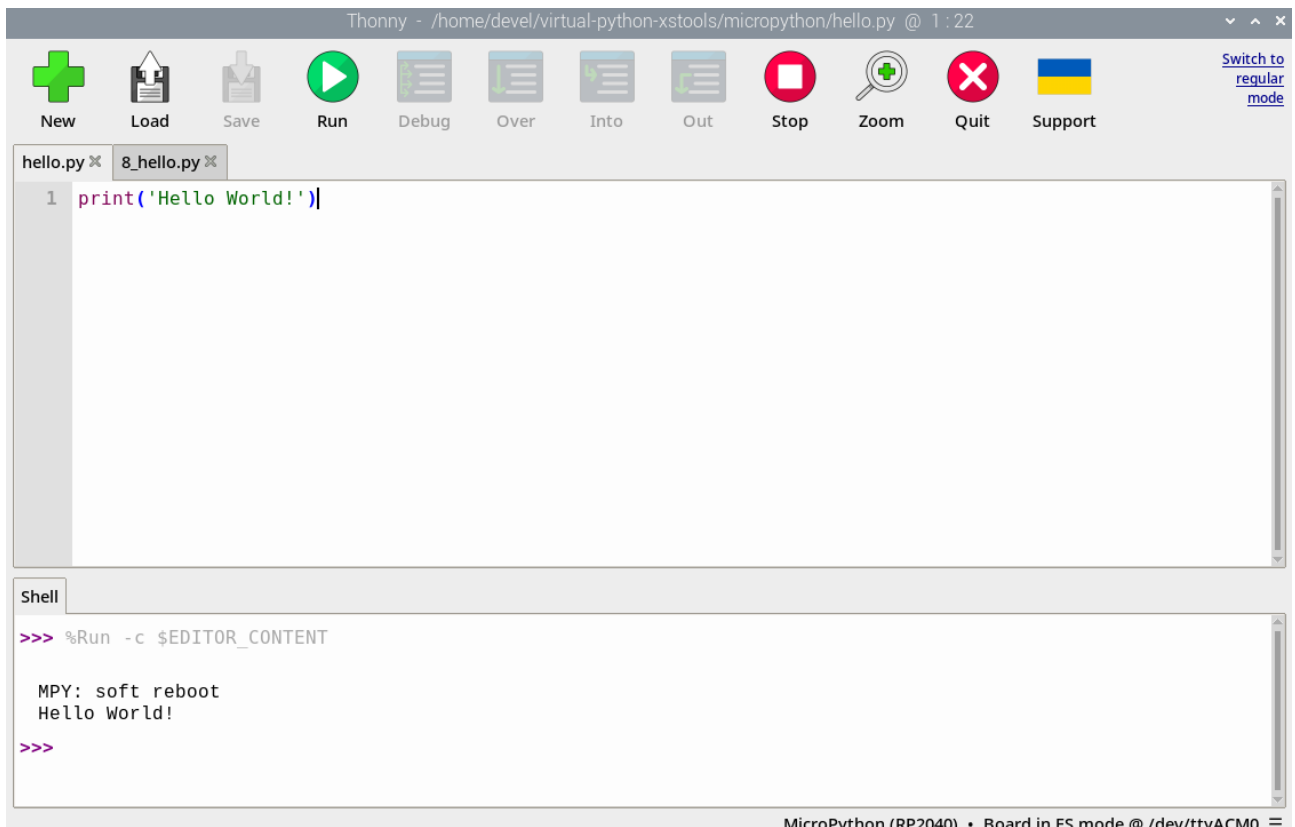
\*\*\*\*\*Default\*\*\*\*\*

Copied “RPI\_PICO-20241025-v1.24.0.uf2” to Raspberry Pico.

Created first MicroPython program hello.py



Running MicroPython program hello.py



The screenshot shows the Thonny IDE interface. The title bar reads "Thonny - /home/devel/virtual-python-xstools/micropython/hello.py @ 1:22". The toolbar contains icons for New, Load, Save, Run, Debug, Over, Into, Out, Stop, Zoom, Quit, and Support. The file explorer shows "hello.py" and "8\_hello.py". The editor window displays the code: `1 print('Hello World!')`. The Shell window shows the command `>>> %Run -c $EDITOR_CONTENT` and the output: `MPY: soft reboot` and `Hello World!`. The status bar at the bottom indicates "MicroPython (RP2040) • Board in FS mode @ /dev/ttyACM0".

```
Thonny - /home/devel/virtual-python-xstools/micropython/hello.py @ 1:22
```

New Load Save Run Debug Over Into Out Stop Zoom Quit Support

hello.py 8\_hello.py

```
1 print('Hello World!')
```

Shell

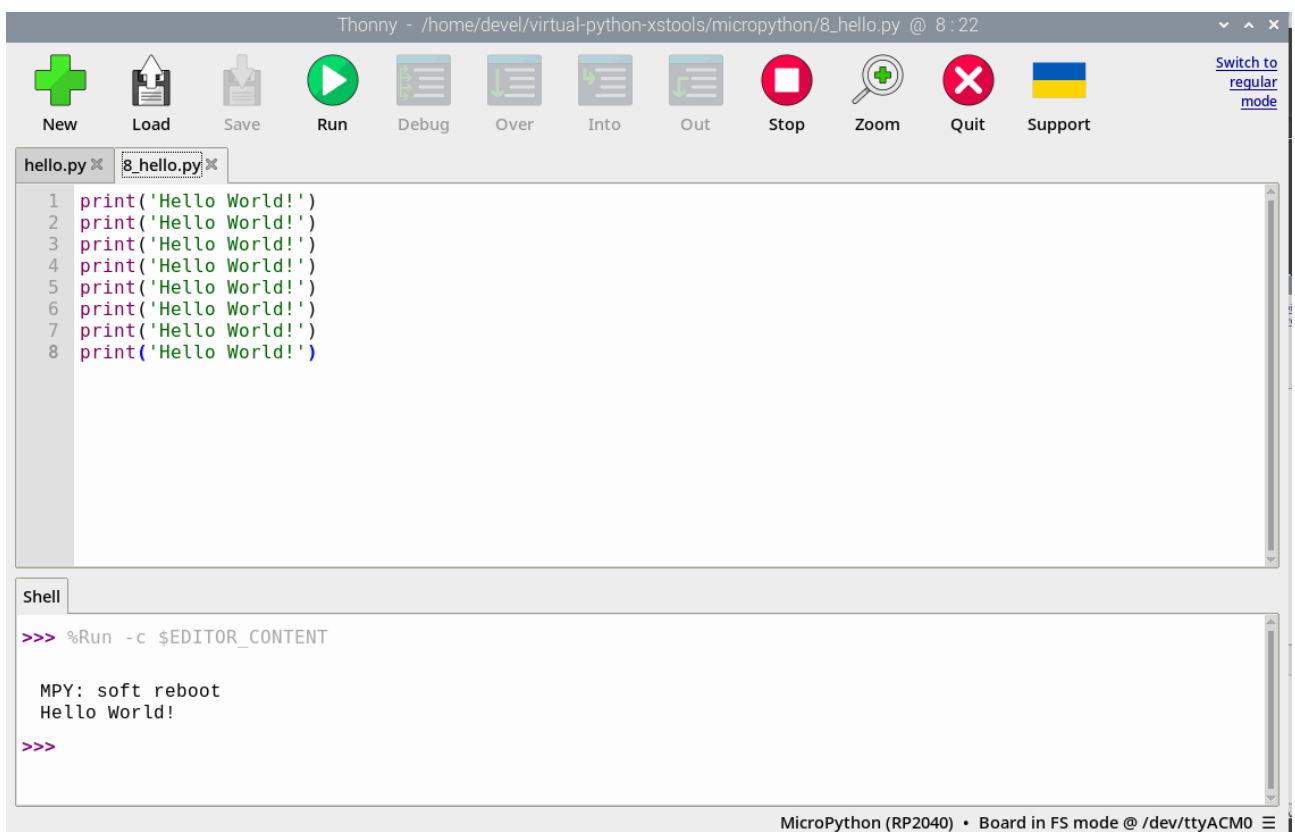
```
>>> %Run -c $EDITOR_CONTENT
```

```
MPY: soft reboot
Hello World!
```

```
>>>
```

MicroPython (RP2040) • Board in FS mode @ /dev/ttyACM0

Modified MicroPython program hello.py 8\_hello.py.



The screenshot shows the Thonny IDE interface. The title bar reads "Thonny - /home/devel/virtual-python-xstools/micropython/8\_hello.py @ 8:22". The toolbar contains icons for New, Load, Save, Run, Debug, Over, Into, Out, Stop, Zoom, Quit, and Support. The file explorer shows "hello.py" and "8\_hello.py". The editor window displays the code: `1 print('Hello World!')`, `2 print('Hello World!')`, `3 print('Hello World!')`, `4 print('Hello World!')`, `5 print('Hello World!')`, `6 print('Hello World!')`, `7 print('Hello World!')`, and `8 print('Hello World!')`. The Shell window shows the command `>>> %Run -c $EDITOR_CONTENT` and the output: `MPY: soft reboot` and `Hello World!`. The status bar at the bottom indicates "MicroPython (RP2040) • Board in FS mode @ /dev/ttyACM0".

```
Thonny - /home/devel/virtual-python-xstools/micropython/8_hello.py @ 8:22
```

New Load Save Run Debug Over Into Out Stop Zoom Quit Support

hello.py 8\_hello.py

```
1 print('Hello World!')
2 print('Hello World!')
3 print('Hello World!')
4 print('Hello World!')
5 print('Hello World!')
6 print('Hello World!')
7 print('Hello World!')
8 print('Hello World!')
```

Shell

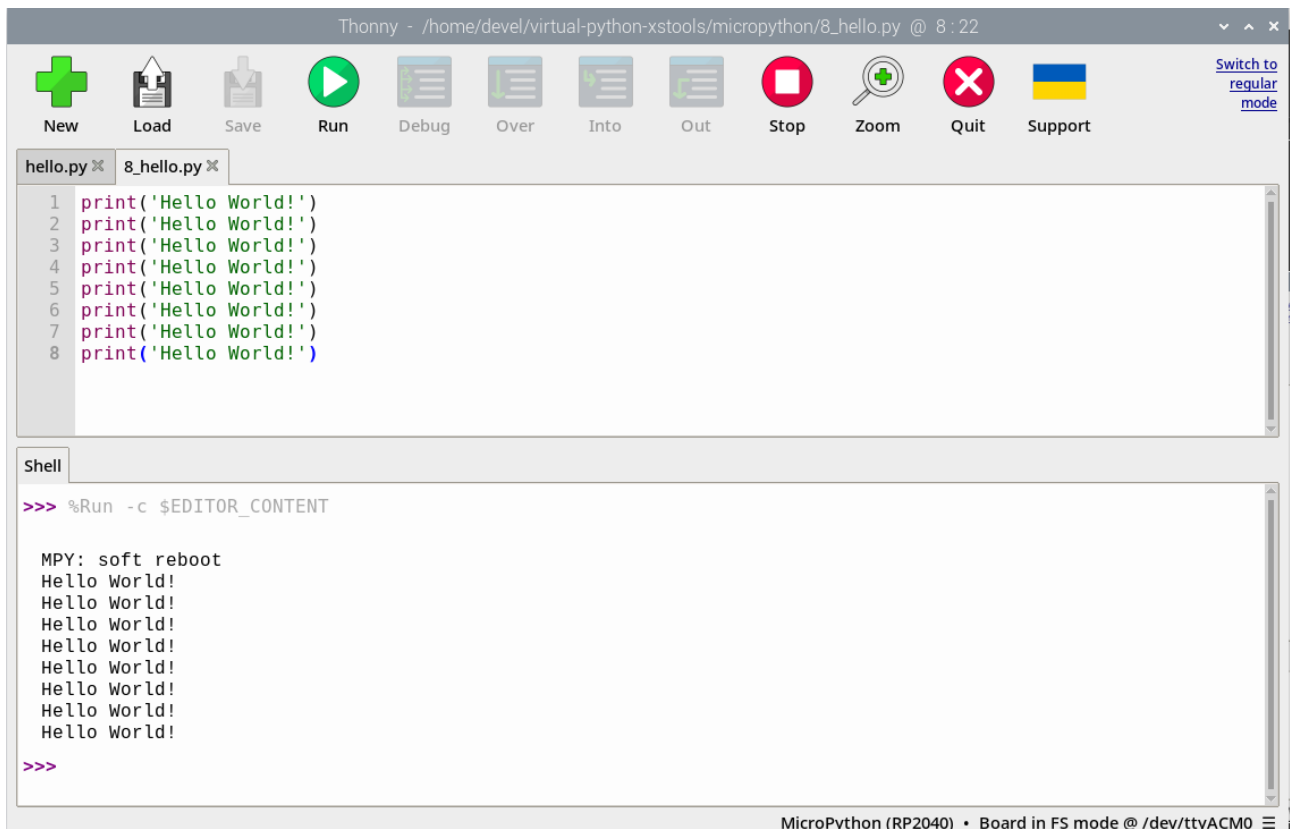
```
>>> %Run -c $EDITOR_CONTENT
```

```
MPY: soft reboot
Hello World!
```

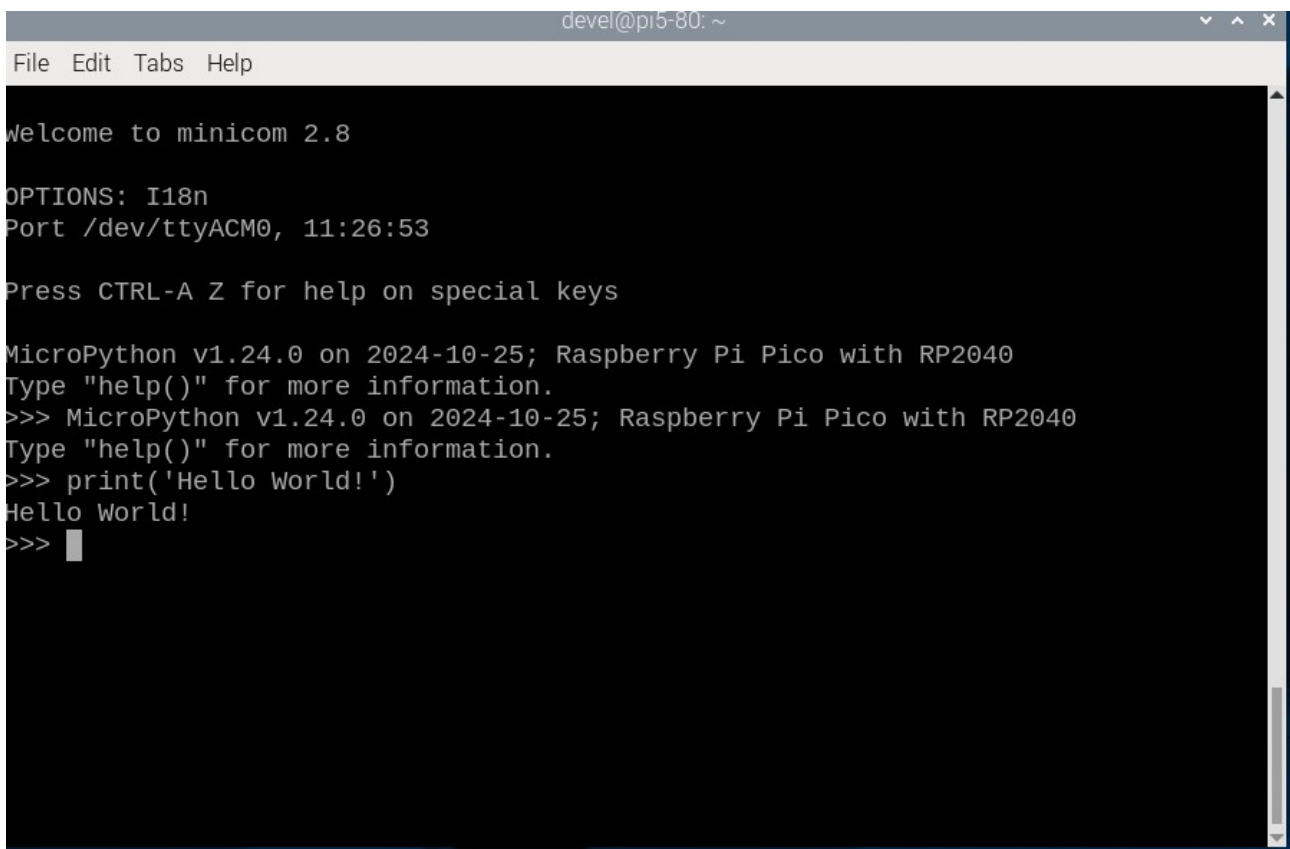
```
>>>
```

MicroPython (RP2040) • Board in FS mode @ /dev/ttyACM0

Running MicroPython 8\_hello.py



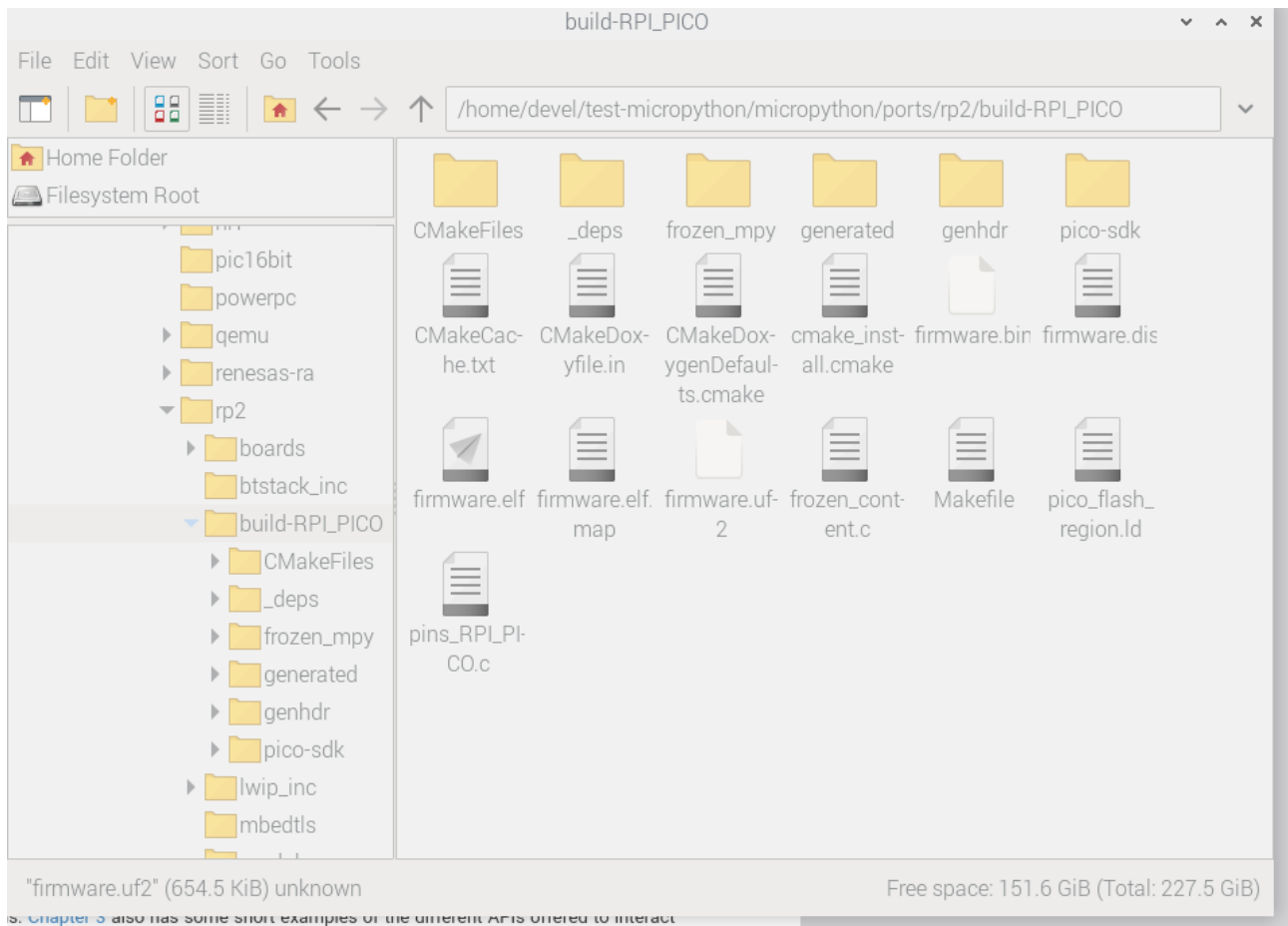
Up to now no imports required.



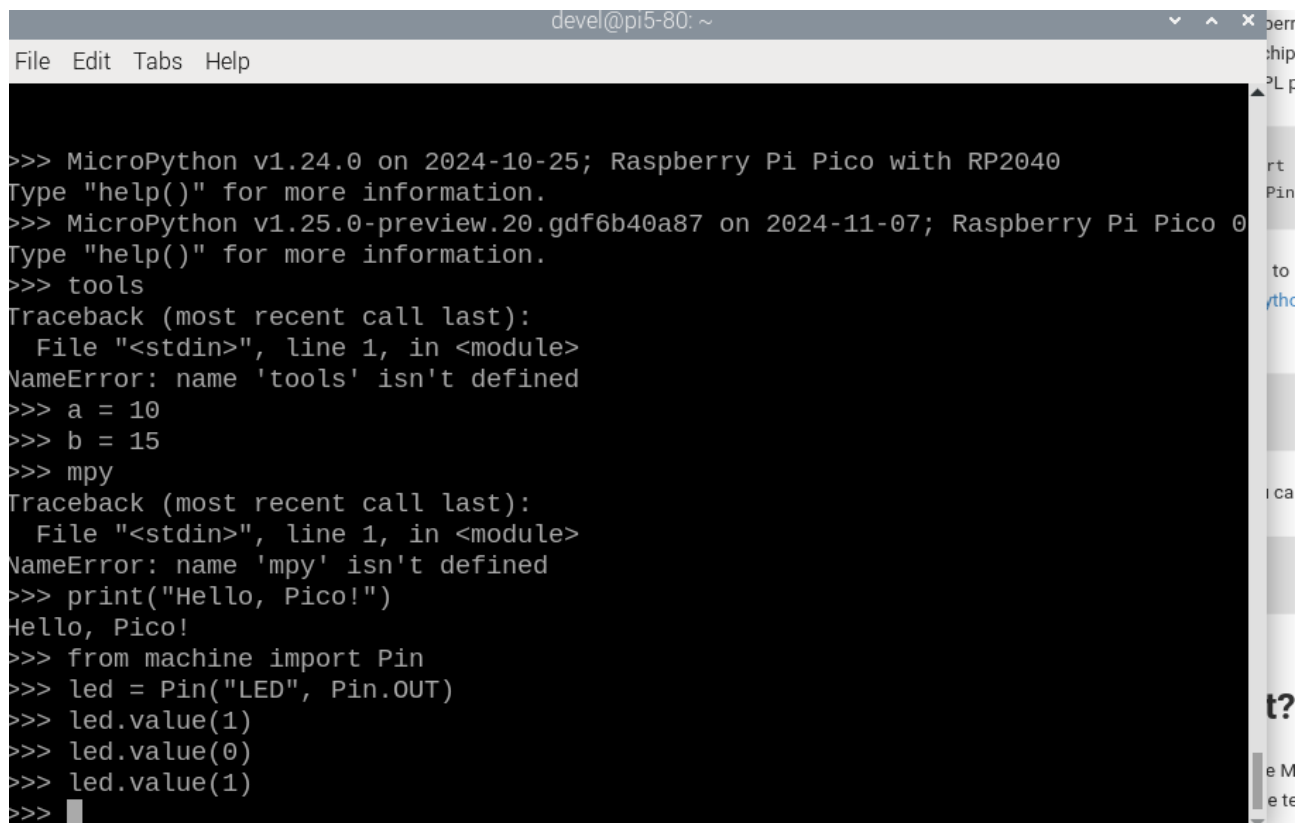
Testing using minicom

```
devel@pi5-80: ~  
File Edit Tabs Help  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>> a = 10  
>>> b = 15  
>>> c = a + b  
>>> d = b/a  
>>> print (d)  
1.5  
>>> print (c)  
25  
>>>
```

MicroPython variables and simple math.



## Built firmware



```
devel@pi5-80: ~
File Edit Tabs Help

>>> MicroPython v1.24.0 on 2024-10-25; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> MicroPython v1.25.0-preview.20.gdf6b40a87 on 2024-11-07; Raspberry Pi Pico 0
Type "help()" for more information.
>>> tools
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'tools' isn't defined
>>> a = 10
>>> b = 15
>>> mpy
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'mpy' isn't defined
>>> print("Hello, Pico!")
Hello, Pico!
>>> from machine import Pin
>>> led = Pin("LED", Pin.OUT)
>>> led.value(1)
>>> led.value(0)
>>> led.value(1)
>>>
```

Led on pico on/off

Need to learn about RPEL

A REPL, or Read-Eval-Print Loop, is :

```
devel@pi5-80: ~
File Edit Tabs Help

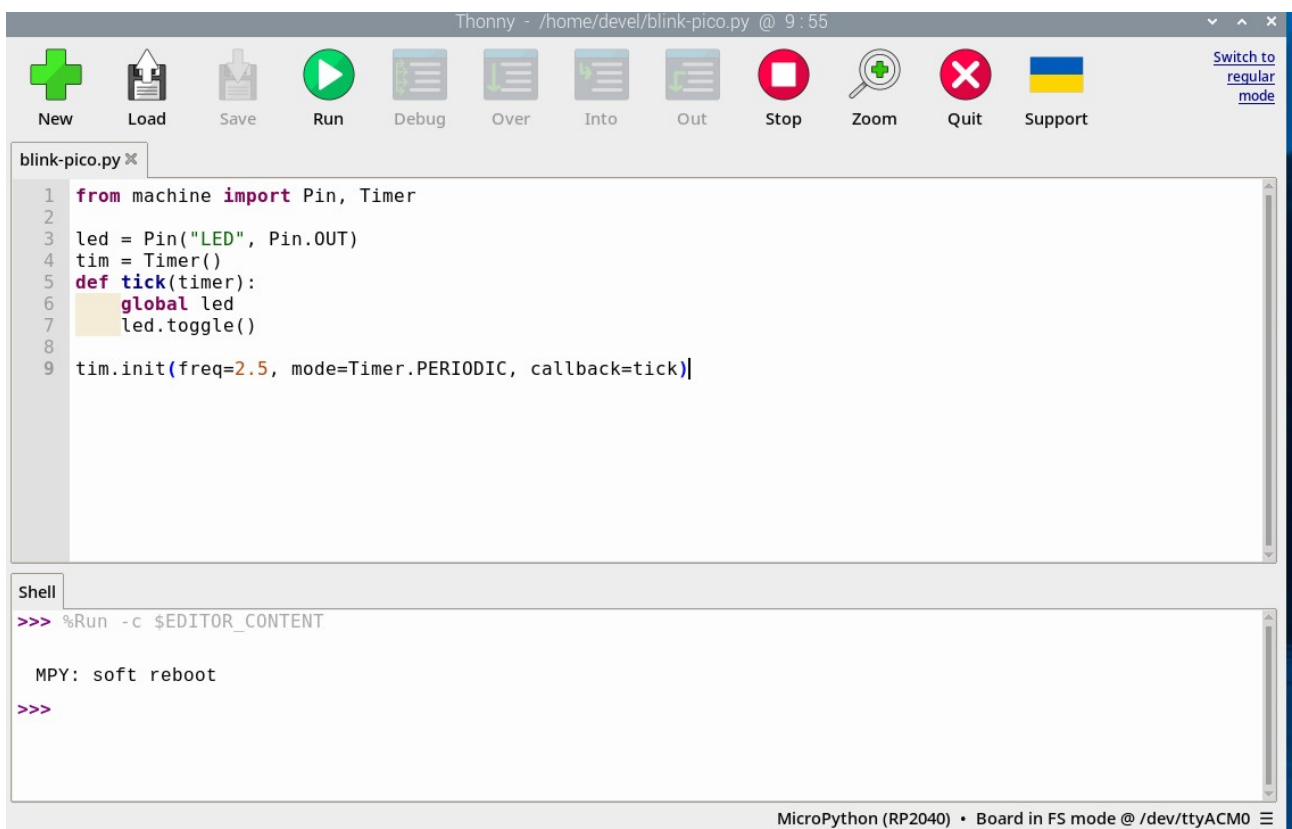
Welcome to minicom 2.8

OPTIONS: I18n
Port /dev/ttyACM0, 18:16:37

Press CTRL-A Z for help on special keys

MicroPython v1.25.0-preview.20.gdf6b40a87 on 2024-11-07; Raspberry Pi Pico with0
Type "help()" for more information.
>>>
>>> from machine import Pin, Timer
>>> led = Pin("LED", Pin.OUT)
>>> tim = Timer()
>>> def tick(timer):
...     global led
...     led.toggle()
...
>>> tim.init(freq=2.5, mode=Timer.PERIODIC, callback=tick)
>>> █
```

Now the led blinks



```
Thonny - /home/devel/blink-pico.py @ 9:55
New Load Save Run Debug Over Into Out Stop Zoom Quit Support
Switch to regular mode

blink-pico.py x
1 from machine import Pin, Timer
2
3 led = Pin("LED", Pin.OUT)
4 tim = Timer()
5 def tick(timer):
6     global led
7     led.toggle()
8
9 tim.init(freq=2.5, mode=Timer.PERIODIC, callback=tick)|

Shell
>>> %Run -c $EDITOR_CONTENT

MPY: soft reboot
>>>

MicroPython (RP2040) • Board in FS mode @ /dev/ttyACM0
```

Thonny now controls pico.