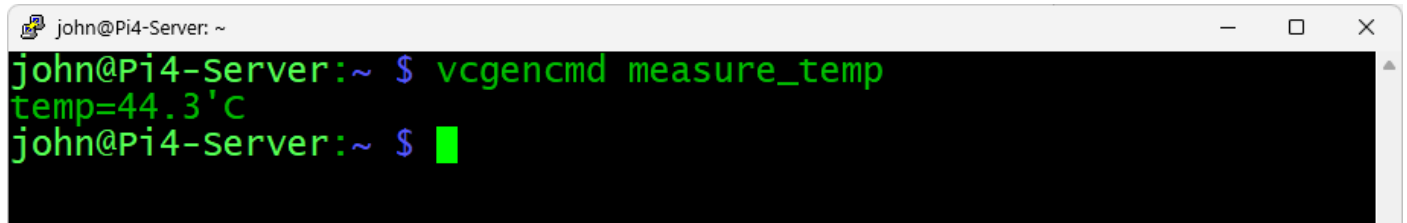


# Hot Pi

One of the problems with a Raspberry Pi is that the temperature could exceed the recommended maximum of 85 degrees Celsius. This maximum is OK for most things, and you would rarely reach it, but if you are overclocking and / or are pushing the CPU you may want to keep an eye on the temperature.

You can check your Pi temperature from the command line using this command:

```
vcgencmd measure_temp
```

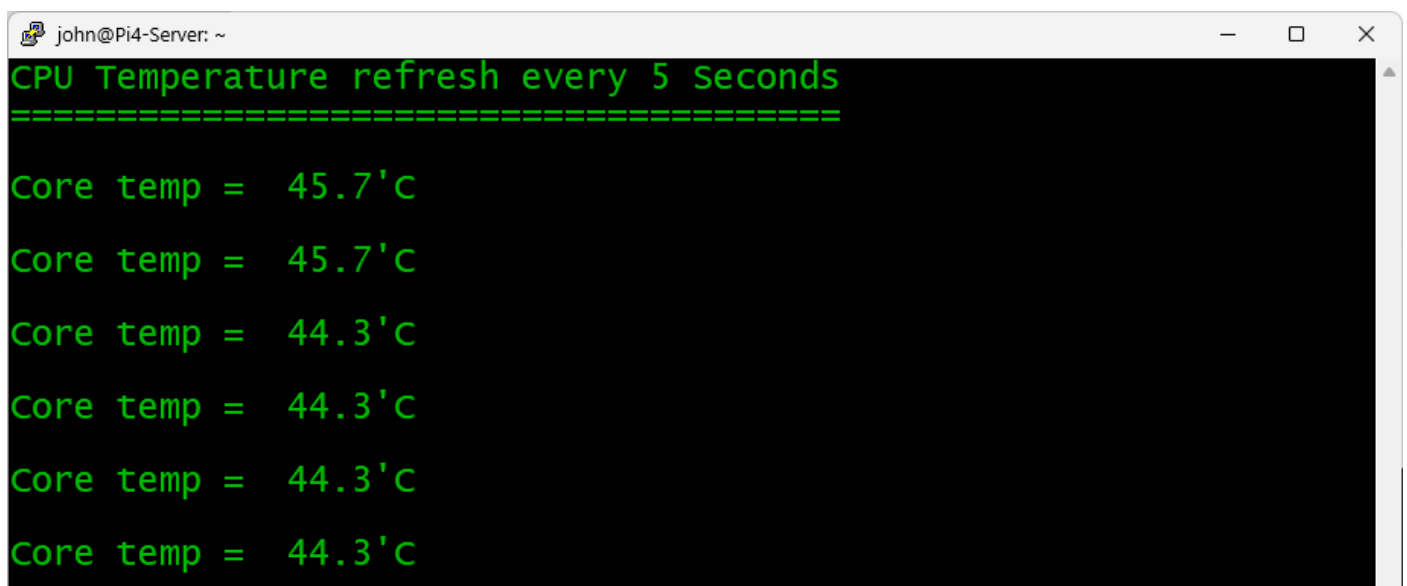


```
john@Pi4-Server: ~  
john@Pi4-Server:~ $ vcgencmd measure_temp  
temp=44.3'C  
john@Pi4-Server:~ $
```

The temperature indicated on my Pi 4B above while it isn't doing much, and this seems to be a reasonably normal operating temperature for a Pi with a heatsink. Heatsinks and Fans are widely available for the Pi and easy to fit should you consider that your Pi needs cooling.

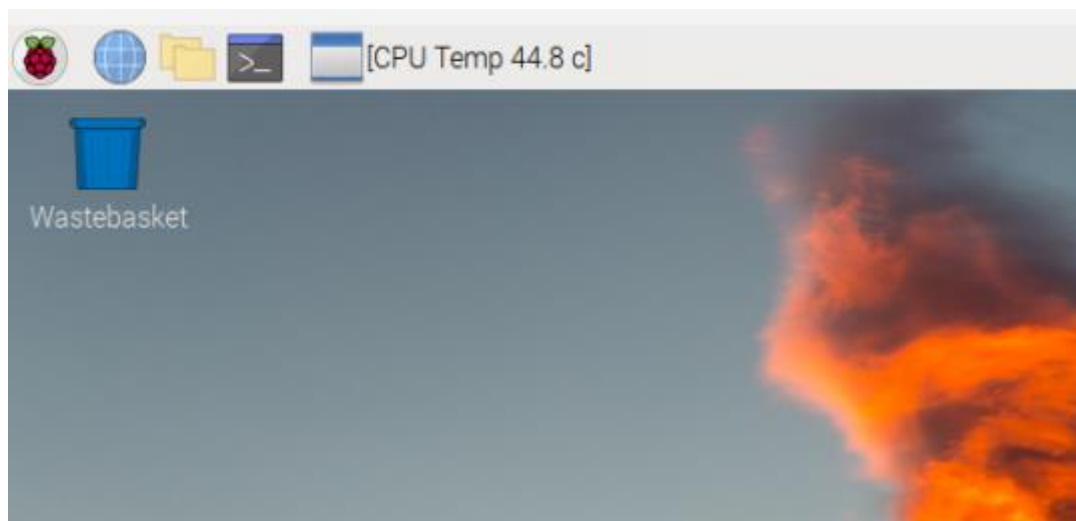
With the command above you will need to check the temperature regularly to see the trend over time. So why not automate the task? The Python code below, when executed in a terminal session, will display the CPU temperature on screen every 5 seconds. (Stop execution with Ctrl Z)

```
import os  
import time  
  
def measure_temp():  
    temp = os.popen("vcgencmd measure_temp").readline()  
    return (temp.replace("temp=", ""))  
  
os.system('clear')  
print("CPU Temperature refresh every 5 Seconds")  
print("=====\n")  
  
while True:  
    print("Core temp = ",measure_temp())
```

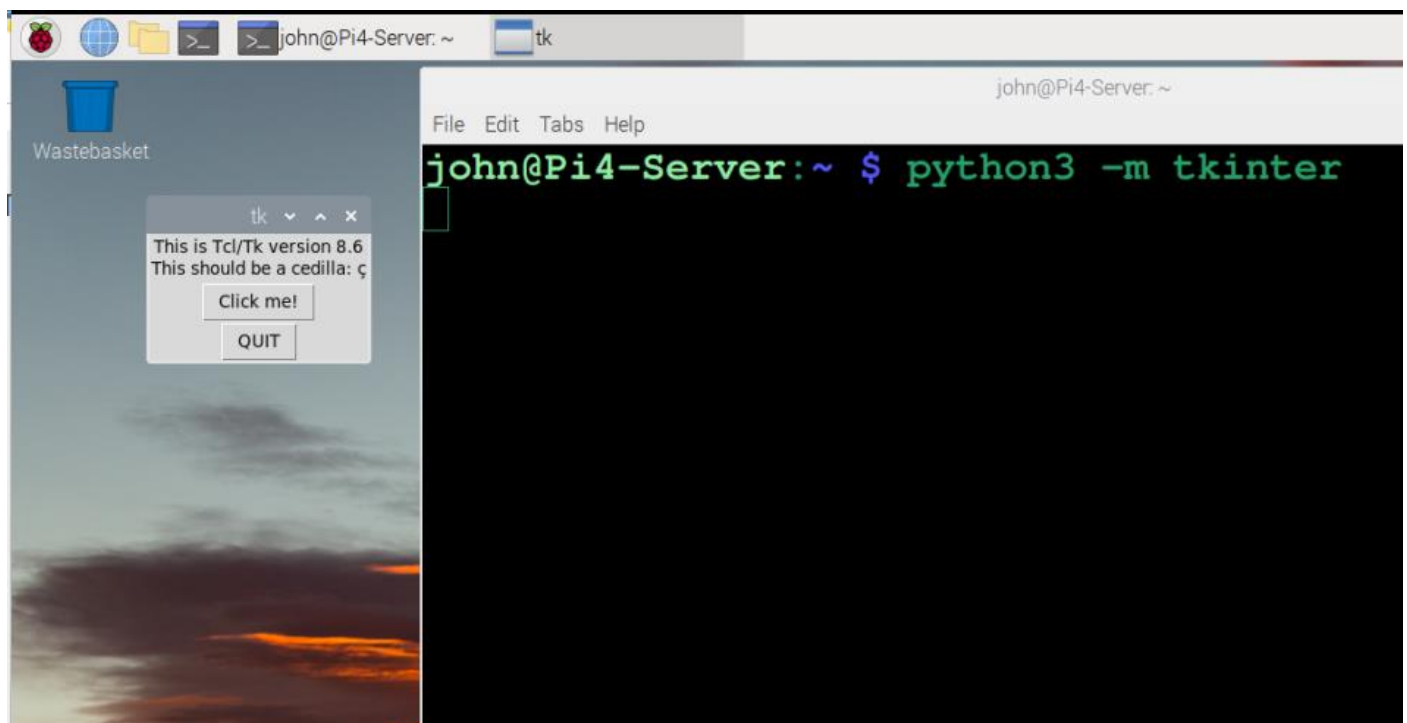


```
john@Pi4-Server: ~  
CPU Temperature refresh every 5 Seconds  
=====  
  
Core temp = 45.7'C  
Core temp = 45.7'C  
Core temp = 44.3'C  
Core temp = 44.3'C  
Core temp = 44.3'C  
Core temp = 44.3'C
```

This allows you to get on with what you are doing and monitor the temperature in another terminal window or by using a terminal emulator, something like PuTTY, if you are on a Windows machine. To monitor the temperature of time, you will need to change the script to write an output to a file. Here I want to see the temperature while using the GUI and put it on the Task Bar as shown below.



You have a choice of program editors in the GUI and the latest Pi OS comes with “tkinter”, which is the standard Python interface and GUI toolkit. Running `python3 -m tkinter` from a terminal window should open a GUI window demonstrating a tkinter and letting you know that tkinter is installed on your system and showing what version.



I am using Geany Programmer's Editor to code the program.

Here is the code.

```
import tkinter as tk
import subprocess

class App:

    def __init__(self, master):

        self.master = master

        self.label = tk.Label(master, text="", width=10)

        self.label.pack()

        self.update_temp()

    def update_temp(self):

        temp = self.get_cpu_temp()

        root.title("CPU Temp "+temp+ " c")

        self.label.config(text=temp)

        self.master.after(5000, self.update_temp)

    def get_cpu_temp(self):

        temp = subprocess.check_output(['vcgencmd', 'measure_temp']).decode('utf-8')

        return temp.replace("temp=", "").replace("'C\\n", "")

root = tk.Tk()

root.geometry("100x30+0+0")

root.wm_attributes("-topmost", 1)

root.resizable(False,False)

app = App(root)

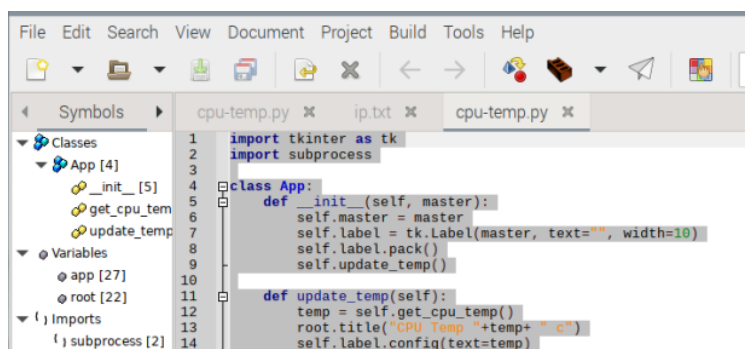
root.mainloop()
```

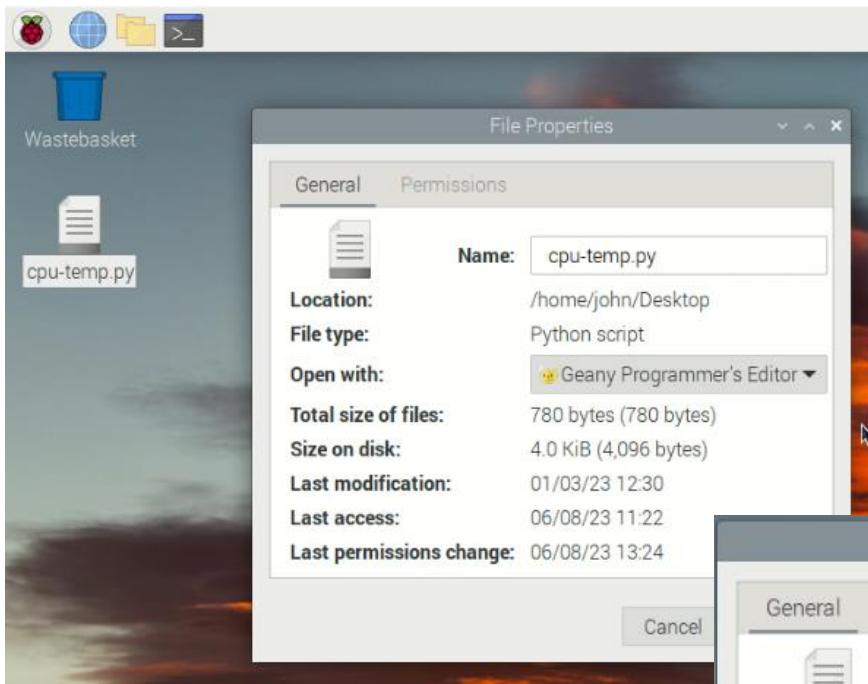
The first routine, “def \_\_init\_\_(self, master)”, defines the interface.

The second, “def update\_temp(self)”, handles the display for updates and calls the third, “def get\_cpu\_temp(self)” which is basically the routine used in the terminal version.

The program can be run from the editor by clicking the “paper plane” icon but it would be nicer to just click the program file to run it.

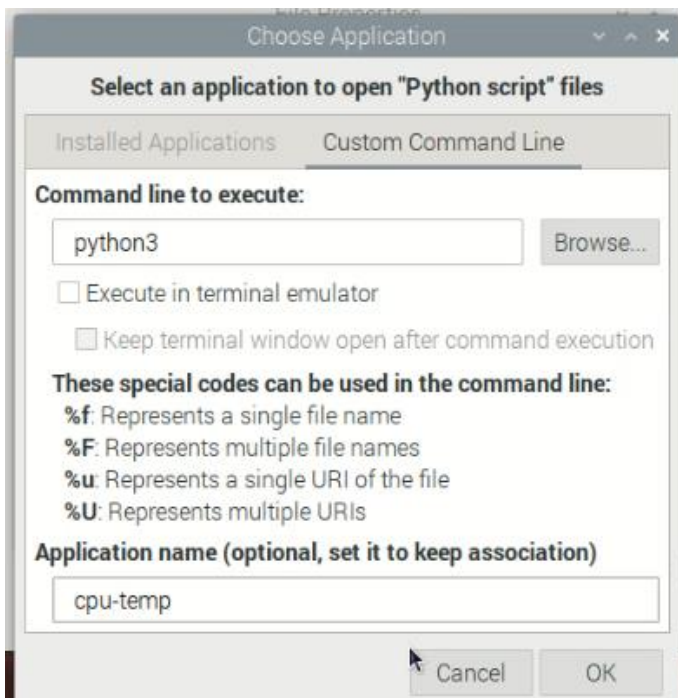
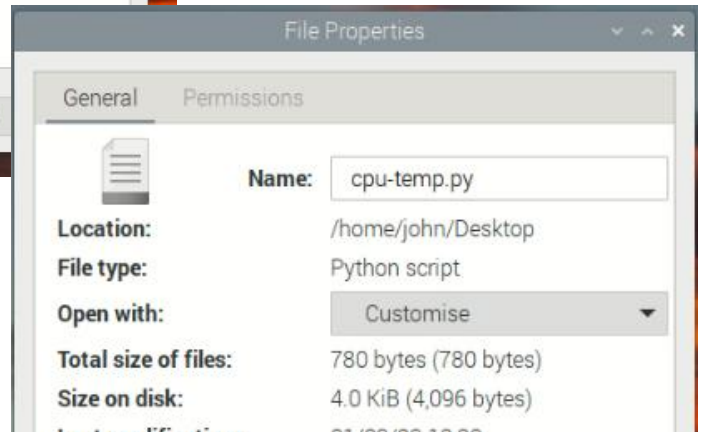
The following show how to achieve this:





Right click the program file and choose properties.

From the Open with: box select customise.



Change Command line to execute to python3.

And here, use the name of the program file without the py extension. And click OK.

Now you can run the program without the editor.

As you see we have a small window with the temperature information but that can be minimised to leave the Task Bar icon which will update every 5 seconds.

