Auto-Running Python Scripts on Raspberry Pi, For Most Purposes Which Python on a Raspberry Pi Might Be Used at Pacific Science Center: A Guide

*Note: `*text within these back-quote marks is code. Don’t include the back-quotes*`*

*```*

Text within triple back-quotes is a multi-line code block.

Don’t include the back-quotes.

*```*

Sources:

[Raspberry Pi: Launch Python Script on Startup : 8 Steps - Instructables](https://www.instructables.com/Raspberry-Pi-Launch-Python-script-on-startup/)

Overview:

Use `crontab`, Linux’s task scheduling tool, to run, every minute, an `sh` file (shell script) that launches your python script.

Within the python script, check if an instance of that script is already running. If it is, quit out of the script.

**Steps**

**Section 0 (optional): put your python script in a folder called `interactive` on the desktop**

For the purposes of this tutorial, I will assume your python script is in a folder called `interactive` on the Desktop, and your Raspberry Pi username is `pi`.

Full folder path: `/home/pi/Desktop/interactive`

**Section 1: Create the launcher shell script (.sh file)**

Open the terminal (command line), the black rectangle icon next to the file explorer icon on the taskbar at the top of the screen.

Navigate to your python script’s directory (which you can find in the navigation bar of the file explorer) by typing:

`cd /home/pi/Desktop/interactive `

Create the launcher file and open it in the nano editor (linux code editor within the terminal) by typing:

`nano launcher.sh`

Within the nano editor, type in this script:

```

#!/bin/sh  
# launcher.sh  
# navigate to home directory, then to this directory, then execute python script, then back home  
  
cd /  
cd /home/pi/Desktop/interactive   
python script\_name.py  
cd /

```

(Where `script\_name.py` is replaced with the filename of your python file.

Ctrl-X, then Return to save and exit.

**Section 2: make the .sh file executable**

Make sure you’re navigated to /home/pi/Desktop/interactive (if not, use `cd /home/pi/Desktop/interactive ` again).

Make the launcher script executable:

`chmod 755 launcher.sh`

Now test it by typing:

`sh launcher.sh`

**Section 3: add method to your python file to check for other instances of the same file**

Add the following imports and method to the beginning of your python script: (source: ChatGPT 😳)

```

import os

import sys

import psutil

def check\_if\_already\_running():

# Get the current script name and PID

current\_process = psutil.Process(os.getpid())

script\_name = os.path.basename(\_\_file\_\_)

# Iterate over all running processes

for proc in psutil.process\_iter(['pid', 'name', 'cmdline']):

try:

# Check if process is a Python process and not the current process itself

if (

proc.info['pid'] != current\_process.pid and

'python' in proc.info['name'].lower() and

script\_name in proc.info['cmdline']

):

print("Another instance of this script is already running. Exiting.")

sys.exit() # Exit the current script if another instance is found

except (psutil.NoSuchProcess, psutil.AccessDenied):

# Ignore processes that have ended or can't be accessed

continue

# Call the function at the beginning of your script

check\_if\_already\_running()

```

**Section 4: use crontab to run launcher.sh every minute**

Crontab is confusing, but it’s pretty powerful. By default it runs tasks in a non-graphical environment, so it won’t display anything on the desktop. You’ll need to specify the display environment.

Check that your display is `:0` by running from the terminal:

`echo $DISPLAY`

If it is `:0`, follow these instructions exactly. Otherwise, use whatever that command returns.

Modify `crontab` for the current user:

In the terminal, type:

`cd /`

`crontab -e`

This will open up the `crontab` editor for the current user. If it prompts you to select an editor, select `nano` (the default option), by just pressing Return.

Arrow-key down to the bottom of the file, and delete anything—if there is anything—below the line that reads:

`# m h dom mon dow command`

Now, at the bottom of the crontab file, type:  
`\* \* \* \* \* DISPLAY=:0 XAUTHORITY=/home/pi/.Xauthority sh /home/pi/Desktop/interactive/launcher.sh`

(Using same path as the launcher.sh file we made in Section 1)

For the audio interactive, additional code is needed, so use the following line:

`\* \* \* \* \* DISPLAY=:0 XDG\_RUNTIME\_DIR=/run/user/$(id -u) XAUTHORITY=/home/pi/.Xauthority sh /home/pi/Desktop/interactive/launcher.sh`

Ctrl-X, `Y`(es), Return to save and exit the crontab editor.

**Section 5: test!**

Wait for the Raspberry Pi’s clock to turn over to the next minute.

Troubleshooting:

1. Go back through each step and make sure there are no typos. Test scripts manually where you can.
2. Reimage the Raspberry Pi’s SD card, and