

0.Preparation before development

0.Preparation before development

1. Close the startup program
 - 1.1. Temporary closing
 - 1.2. Permanent shutdown
 - 1.3. Set up auto-start at boot
 - 1.4. Temporary start
2. Start Jupyter Lab
3. HelloWorld
 - [Code Block](#)
 - [Write code](#)
 - [Run code](#)
 - [Code block description](#)
 - [Notes](#)

1. Close the startup program

In order to facilitate the experience of the APP function of the car, a program is added to the system. This program combines the control functions and gameplay of the APP, so it is called "big program". When the motherboard system is turned on, this program will automatically start, so it is called "boot-start big program".

The startup self-start big program is just for the convenience of experiencing the mobile phone APP function. In actual development, it is necessary to close the big program, otherwise it will occupy the device and cause unpredictable errors. Therefore, before developing the program, please manually close the startup self-start big program.

There are two ways to close the startup self-start big program. One is to close it only once, and it will automatically start the next time you start it. It is called temporary closing. The other way is that after closing it, it will not automatically start again the next time you start it, unless it is manually reopened, which is called permanent closing.

1.1. Temporary closing

To close it temporarily, you need to log in to the Raspberry Pi remotely first. For information on how to log in, please refer to the section **4. Remote Access** in "**11. Development Environment Setup**"

After logging in remotely, enter the following command in the terminal to temporarily shut down

```
sh /home/pi/project_demo/raspbots/killprocess.sh
```

1.2. Permanent shutdown

After logging in remotely, enter the following command in the terminal to permanently shut down

```
rm -rf /home/pi/project_demo/raspbots/start_raspbots.desktop
```

1.3. Set up auto-start at boot

After logging in remotely, enter the following command in the terminal to temporarily start

```
cp -r /home/pi/project_demo/raspbots/start_raspbots.desktop  
/home/pi/.config/autostart/
```

1.4. Temporary start

Before temporarily starting a large program, you need to check whether there is a process occupying the network port

```
lsof -i :6001
```

Note: The value of the kill command PID (892808) needs to be modified according to the actual terminal display, otherwise the port occupation cannot be closed

```
kill 892708
```

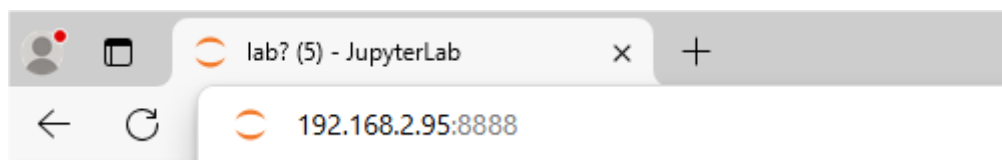
```
pi@yahboom:~$ lsof -i :6001
COMMAND  PID USER  FD  TYPE  DEVICE  SIZE/OFF  NODE NAME
python3  892708 pi    8u   IPv4  2173090      0t0  TCP *:x11-1 (LISTEN)
python3  892708 pi   10u   IPv4  2173090      0t0  TCP *:x11-1 (LISTEN)
pi@yahboom:~$ kill 892708
pi@yahboom:~$
```

```
python3 /home/pi/project_demo/raspbots/raspbots.pyc
```

2. Start Jupyter Lab

Note: The factory image has been set to start automatically, no need to run the command again

Example: The IP of the Raspberry Pi is 192.168.2.95. After starting Jupyter Lab, you can enter 192.168.2.95:8888 in the browser of the device in the same LAN or the local machine to access Jupyter Lab. If prompted for a password, enter: yahboom

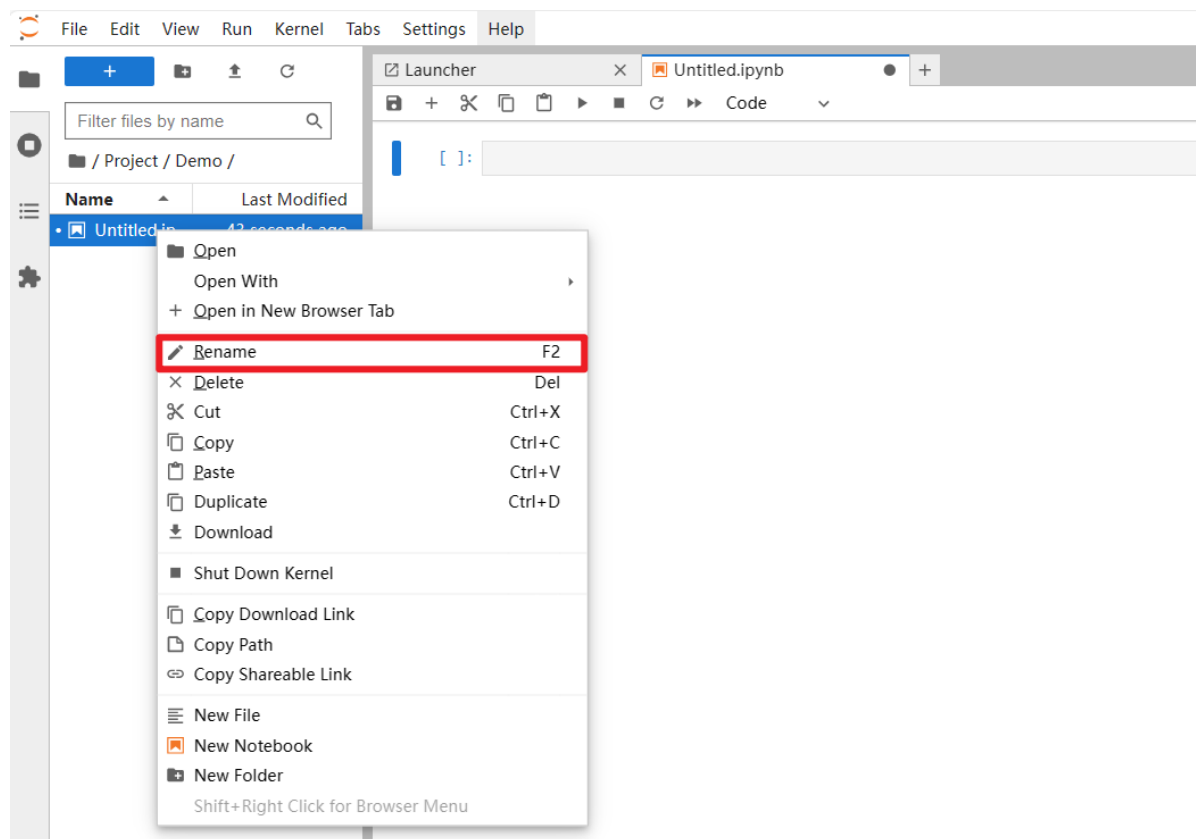
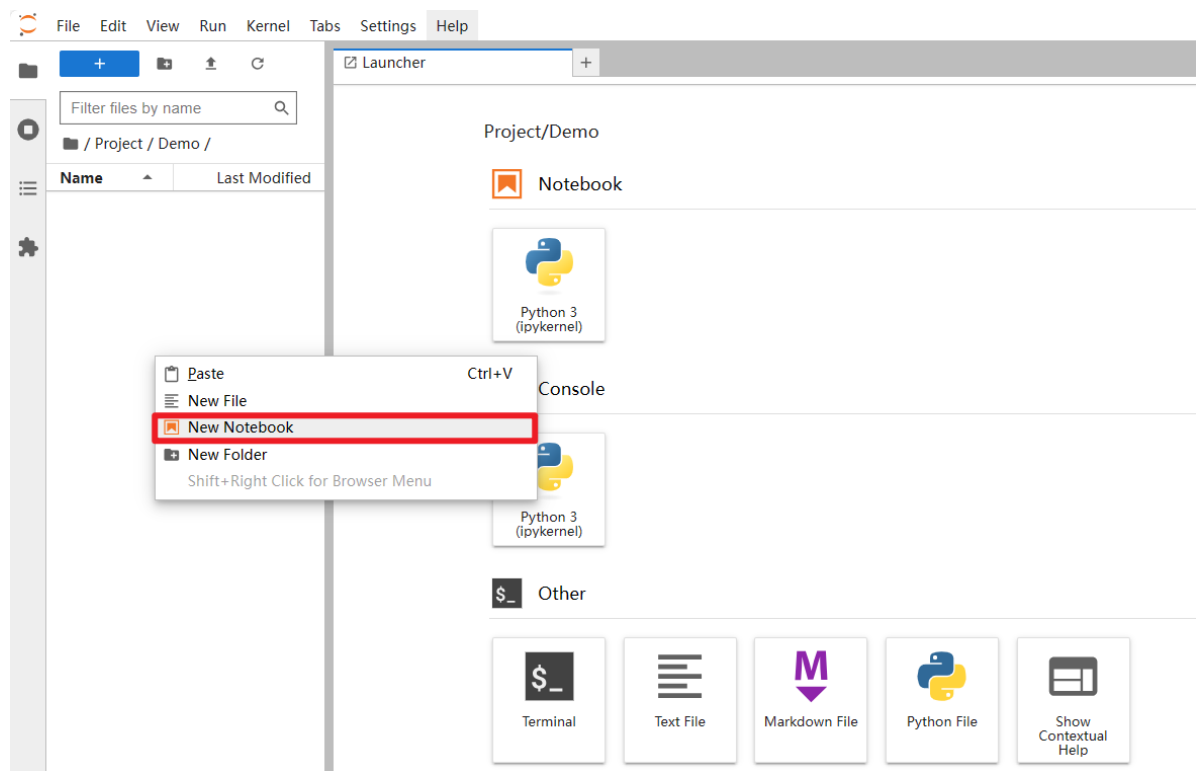


3. HelloWorld

Write and run the HelloWorld program.

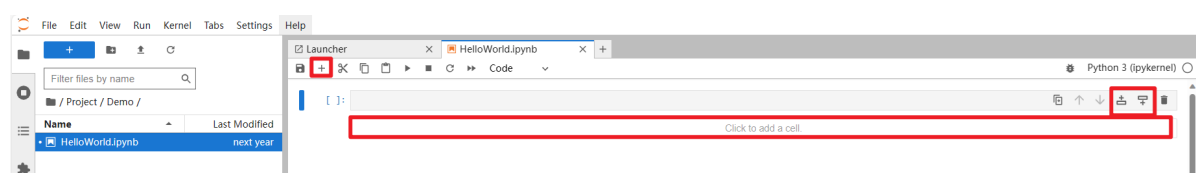
- Create a new file and name it HelloWorld

Right-click the file browser menu bar, select "New Notebook", select Python3 as the kernel, and rename it to HelloWorld

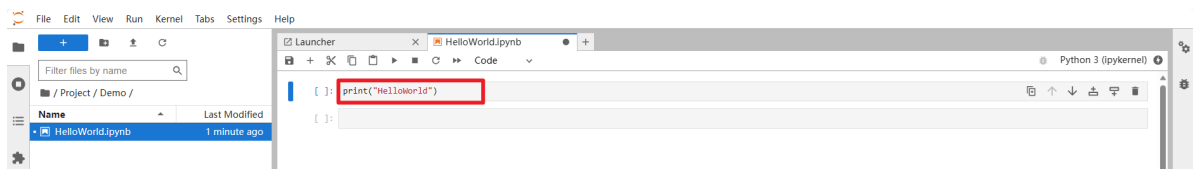


Code Block

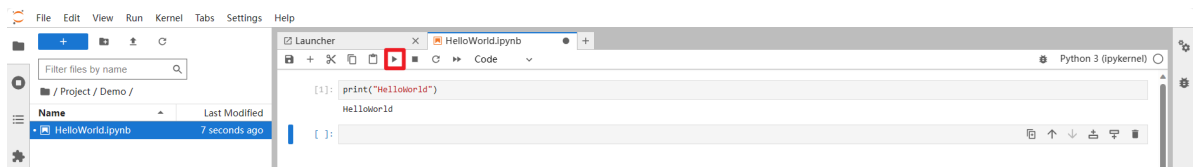
Select the following position to add code blocks up or down



Write code



Run code

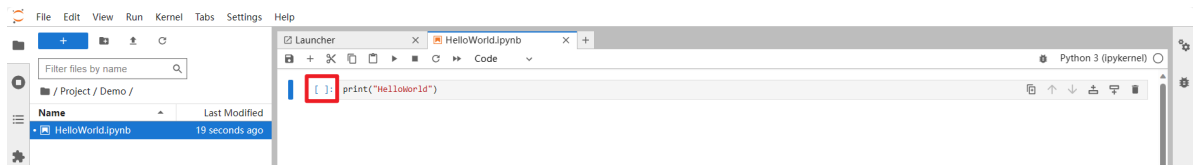


Code block description

The content in the [] on the left side of the code block has two states: * and numbers

*: Indicates that the code block is still running

Numbers: Indicates the order in which the code block is executed



Notes

If you want to reset or encounter abnormal problems during the code running, you can follow the following steps: restart all kernels and clear all code block outputs.

