NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

Exercise Manual
for
SC1003
Introduction to Computational Thinking and Programming

Exercise #1.1:
Introduction to Raspberry Pi

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY

Learning Objectives

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

Exercise Manual
for
SC1003
Introduction to Computational Thinking and Programming

Exercise #1.1:
Introduction to Raspberry Pi

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY

Learning Objectives

In this course, students will learn the various concepts of computational thinking through programming exercises performed on the Raspberry Pi platform. This laboratory exercise is to introduce students to the Raspberry Pi (RPi) single-board computer, and how to use its text-based (Linux's bash) commands to explore the environment of the RPi.

Intended Learning Outcomes

At the end of this exercise, you should be able to

- Operate the Raspberry Pi board through its Graphic User Interface and through its Terminal mode interface.
- Write basic programs in Python language and C language to observe the difference in running an interpreted language and a compiled language based programs.

Equipment and accessories required

i) Raspberry Pi 3 Model B (RPi3) board with Sense HAT add-on display module/board.

ii) PC or notebook

---

1. Introduction

The Raspberry Pi (RPi) is a small single-board computer developed in UK by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools. It is a low cost computer board (about S$50) but comes with powerful processor (that are also used in some of the latest smartphones).

Fig.1 - Raspberry Pi 3 Model B (RPi3)

For the RPi3 model that is used in this exercise, the board also support two wireless interfaces (i.e. connections), Wifi and Bluetooth, which means that it can also be accessed using your wireless devices such as Notebook and Smartphone.

As an educational kit, you can add other devices to the RPi board, typically in the form of add-on modules. In this course, you will be using the **Sense Hat** add-on module on the RPi board, such that you can code programs to display messages and images on the module, as will be seen later.

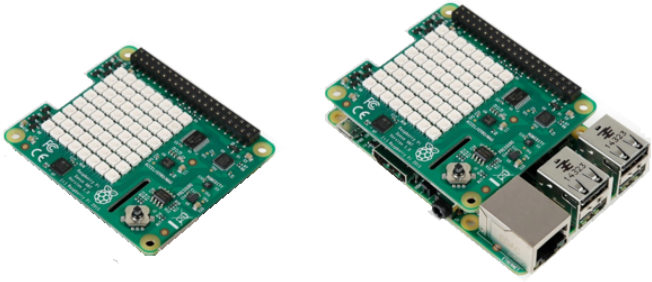


Fig. 2 - Sense Hat module and its interface on RPi

The RPi can be set up as a standalone computer by connecting a monitor, keyboard and mouse through its various interfaces.

Fig. 3 - Standalone setup of RPi

The operating system (OS) used on the RPi is the **Raspbian OS**, which is based on the open source Linux OS. (An OS is the software used to control and operate a computer, such as the Microsoft's Windows OS and Apple's macOS for their notebooks, and Google's Android and Apple's iOS for their smartphones)

1a Accessing the RPi board by remote access method

While the RPi can be used as a standalone computer by connecting it with keyboard/mouse/monitor as in Fig 3, this setup is rather cumbersome due to the connections required. An alternative way to use the RPi is to operate it in a 'headless' setup, i.e. **without connecting it with keyboard/mouse/monitor**. Instead, it is connected via one of its network interfaces to a computer/notebook. In this course, the setup is to have the RPi connects to the desktop computer in the laboratory using its wired Ethernet interface as shown in Figure 1 (and powered by one of the USB port on the computer).

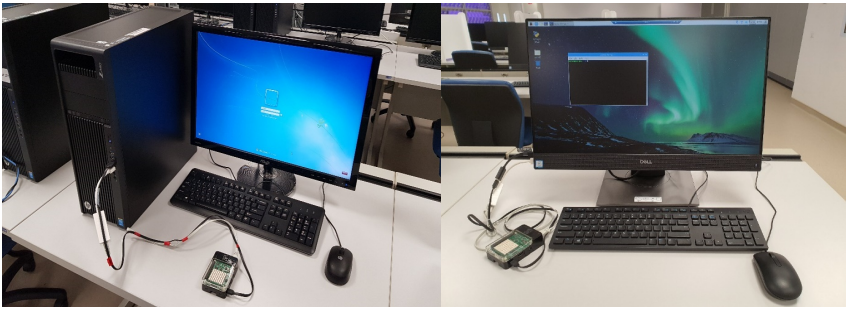


Fig.3a - Raspberry Pi in Headless mode

Connect Rpi board to PC

To access and use the RPi board, you have to **connect the Rpi to 2 USB ports on the PC** as shown in Fig 3a above.

Launch “Remote Desktop Connection” program and follow the steps below :

1 Launch “Remote Desktop Connection” from PC desktop

Use IP address **11.11.11.11**

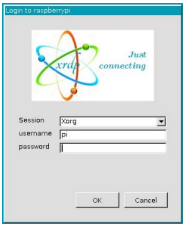
2 Click “Yes” on the Identity pop up window

Remark :
The Rpi board has a pre-set IP address of 11.11.11.11

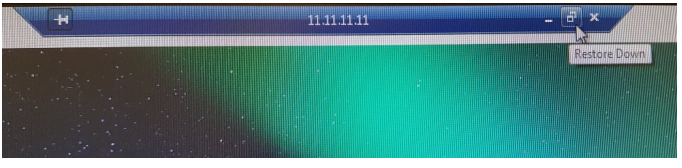
Use the following to login to Pi :

3 Session : Xorg

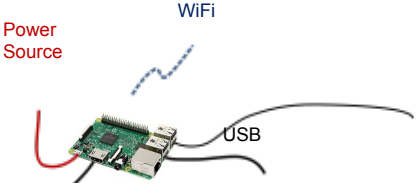
Username : pi
Password : raspberry

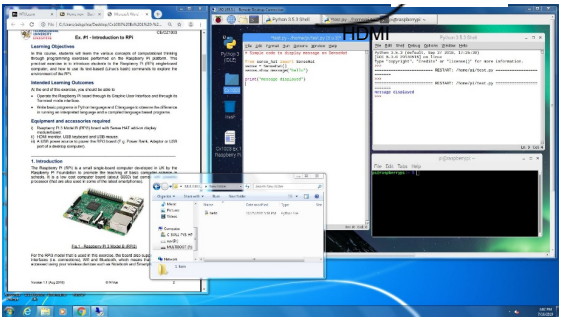


Click on the Resize Button on the Top blue bar to resize the remote desktop **for ease of accessing the PC and remote session together**



4





Remarks :

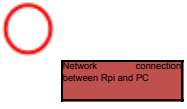
You could use "right click" copy and paste between PC and the "RPi remote desktop window" for easy file transfer and even copy/paste codes

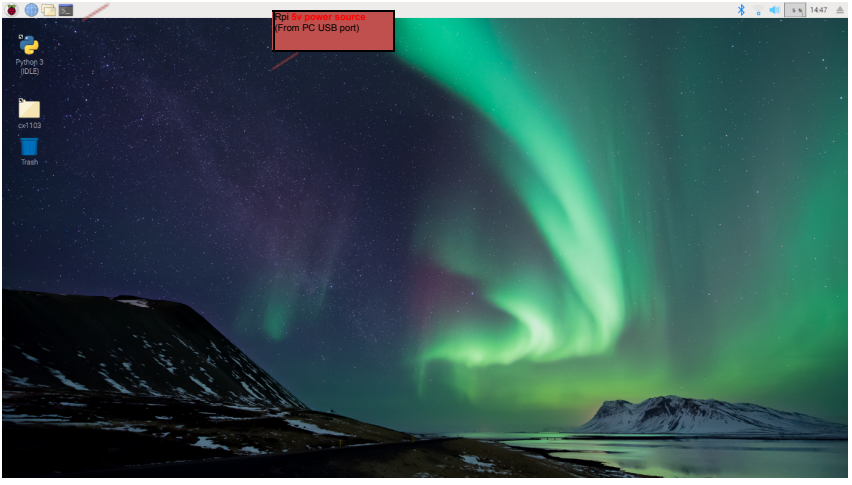
Note : right click copy/paste will not work on folder

2 Familiarization of the RPi's User Interface and useful commands

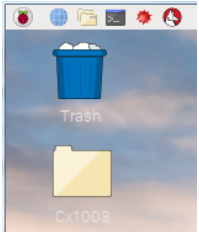
You will now go through a sequence of tasks designed to get you familiarized with the operation of the RPi, in particular, certain commands that are frequently used on RPi.

2.1 This is the Graphic User Interface (GUI) based Desktop provided by Raspbian.





2.2 There are several ‘icons’ available on the Desktop that allow you to interact and access the various functions available on the RPi3 system.

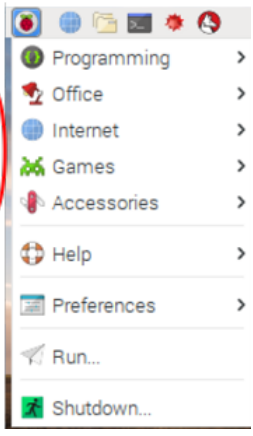


- Click on the **Application Menu** icon - you will be provided with a list of options - programs and functions available on RPi, which you can use to operate the RPi. Note that there is a **Shutdown** function which is used to properly turn off the RPi at the end of the exercise.

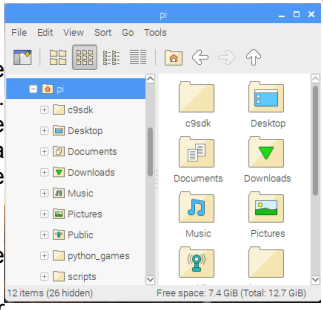
- The **Web Browser** is normally used for surfing internet, which you will use as needed during the various hands-on exercises in this course.

- **File Manager** is used to explore the directories and files stores on the RPi.

- **Terminal** is a program that is usually used to enter text commands and run other programs, as will be seen later.



2.3 Click on the **File Manager** icon. You will be presented with a window showing the various folders/directories and files available on the RPi.

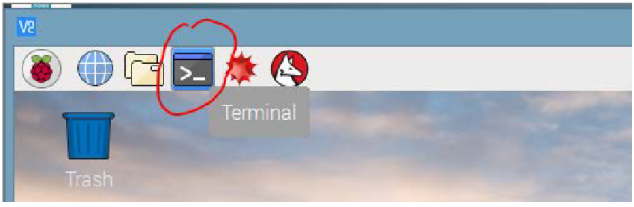


- You can further explore the various folders by clicking on each of them to see their contents.
- Click the **'Desktop'** folder and explore its content.

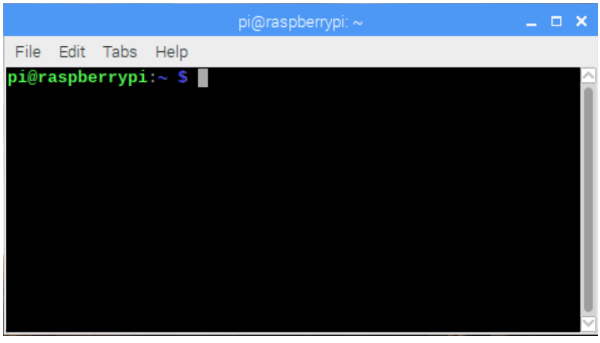
3. Finding information using commands in Terminal mode

While the GUI based Desktop is convenient and user friendly, there are other things that you can do with RPi, many of which can only be accessed by typing text-based commands through the Terminal. In practice, using Terminal mode is the most 'powerful', flexible and quickest way to accomplish many things on RPi.

3.1 Click on the **Terminal** icon to open a terminal window.

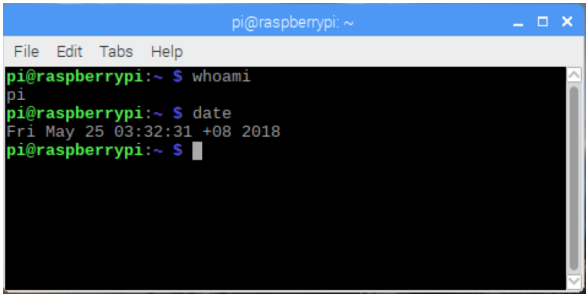


• Resize the terminal window so it's of a reasonable size for you to type into.



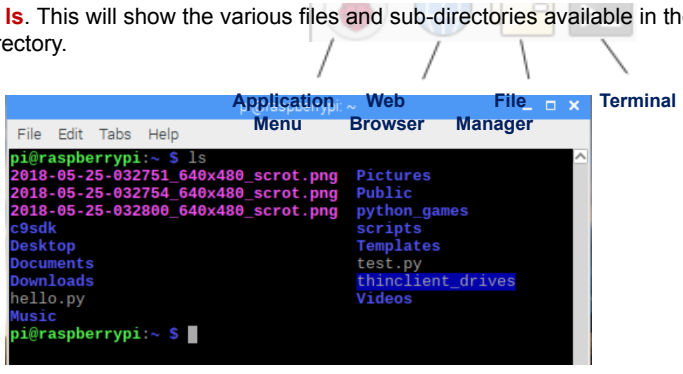
3.2 Type the following commands into the terminal and observe what is shown for each of them.

- **whoami**
- **date**

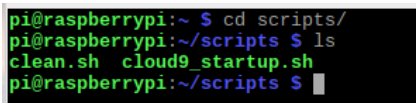


3.3 You can view the contents in the current directory by using the list

command **ls**. This will show the various files and sub-directories available in the current directory.



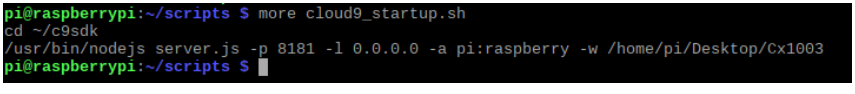
3.4 You can change to a different directory, such as to the **scripts** directory using the command: **cd scripts**



- View the contents of this directory (using the **ls** command again).

3.5 You can view the content of the file by using the command **more**

- Type the command **more cloud9_startup.sh**
- The content of this file is then shown on the screen.



Helpful tip: The <Tab> key is very useful when typing command in the terminal mode, such as to select a file in a directory without entering the full name of the file. E.g., you can execute the full command by merely typing **more clo**<Tab>. Try it.

3.6 You can check what directory you are currently working in

- Type the command **pwd**

3.7 You can change to the directory above the current directory as follows

• Type the command **`cd ..`**

There are many other commands and things that you can do in terminal mode, which you will learn in due course, but we will now move on to do a bit of programming exercises on the RPi board.

4. Interpreted language versus Compiled language

You have learnt in Topic 1 of the recorded video about the different type of programming languages, which can be further separated into interpreted language, such as Python and compiled language such as C. In this exercise you are going to write a simple program using each of these two languages and observe how each of them get executed.

4.1 Python is an 'interpreted language'. You write the code and run the program. There are two ways to run a python code, the Script mode and the Interactive mode. You will use the Interactive mode here which is clearer for illustrating the working of an interpreter.

• Open a terminal window in RPi and launch the Python **interpreter, python3** by entering

`python3`

• You will see the prompt **>>>** display on screen

```
pi@raspberrypi:~ $ python3
Python 3.5.3 (default, Jan 19 2017, 14:11:04)
[GCC 6.3.0 20170124] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

• Key in the following Python code, and press *Enter*

```
print ("Hello world")
```

You will see that the message is shown on the screen immediately in the next line.

```
pi@raspberrypi:~ $ python3
Python 3.5.3 (default, Jan 19 2017, 14:11:04)
[GCC 6.3.0 20170124] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print ("Hello world")
Hello world
>>>
```

And that is – the code that you entered has been executed by the interpreter python3.

• Next key in the following and press *Enter*

```
1+2
```

You will notice that the code is immediately executed by the interpreter as soon as you press the *Enter* key.

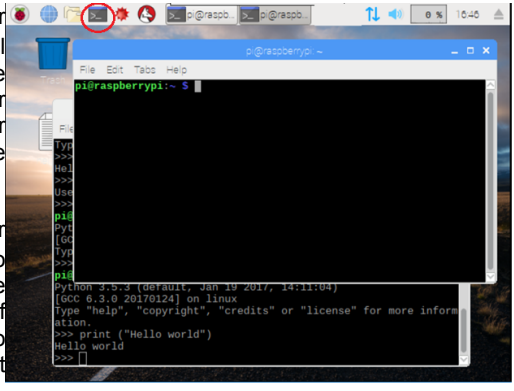
```
pi@raspberrypi:~ $ python3
Python 3.5.3 (default, Jan 19 2017, 14:11:04)
[GCC 6.3.0 20170124] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print ("Hello world")
Hello world
>>> 1+2
3
>>>
```

(In practice, we will usually use the **Script mode** as described in Appendix A, which is a more practical way to develop a python program)

We will open another Terminal here. (This is to also demonstrate the multi-tasking ability of the RPi, i.e. its ability to run multiple programs at the same time).

4.2 Let us now write a program in C to do exactly the same thing. Unlike the Python program, you need to first create a file to key in the program code in C. In here we will use a text editor program called **nano**.

• To do so, you can either use the existing Terminal (i.e. the one still runs the Python3 interpreter), or you can open another terminal by clicking the Terminal icon.



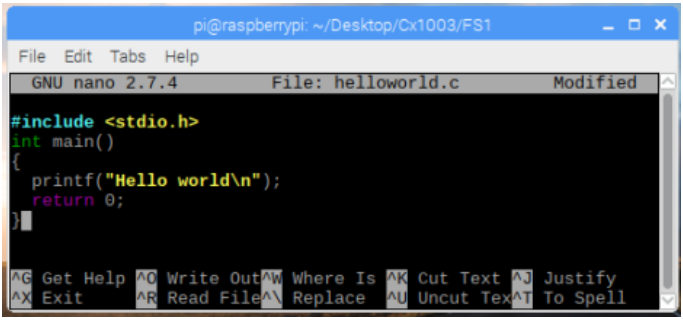
• To do so, you can either use the existing Terminal (i.e. the one still runs the Python3 interpreter), or you can open another terminal by clicking the **Terminal** icon.

• We will open another Terminal here. (This is to also demonstrate the multi-tasking ability of the RPi, i.e. its ability to run multiple programs at the same time).

• First, change to the subdirectory of your lab group by typing **cd ~/sc1003/xxx** (replace xxx with your lab group, eg sc1). Then use the text editor program, **nano** to create a file calls **`helloworld.c`** using the command as shown below

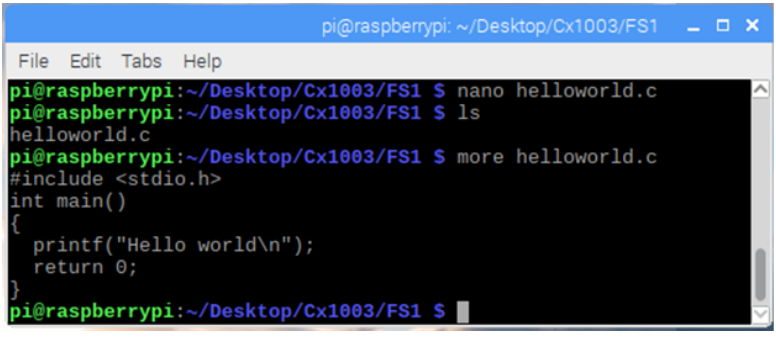
```
pi@raspberrypi: ~/Desktop/Cx1003/FS1
File  Edit  Tabs  Help
pi@raspberrypi:~ $ ls
c9sdk     Documents  Music     Public        scripts     thinclient_drives
Desktop   Downloads  Pictures  python_games  Templates   Videos
pi@raspberrypi:~ $ cd Desktop/Cx1003/FS1
pi@raspberrypi:~/Desktop/Cx1003/FS1 $ nano helloworld.c
```

- Key in the program code as shown below



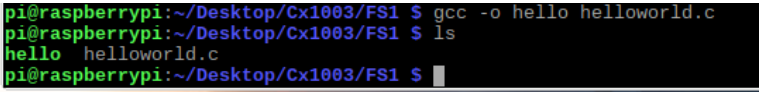
- Save the program and close the **nano** text editor by pressing the keys Ctrl-X (i.e. *Ctrl* key together with the *X* key), type "y" when prompted to 'Save modified buffer?', press [Enter] key to confirm save.

- What you have done so far is you have created a C program file. You can check the existence of this file and see its contents using the commands that you have learnt earlier.



- In order to execute this C program, you will need to first compile it. To do so, you will need to run a compiler, and here you will use the **gcc compiler**.

- Type the gcc command as shown below. If everything is OK, then a new file **`hello`** will be created as shown - this is the executable file created by the compiler.



- If instead, you see other message(s) and cannot find the hello file, it means

that you probably have made a mistake somewhere, either in your code or in the command that you typed. This is then a great opportunity for you to learn about 'debugging' – to find out what is/are wrong and fix the problem(s).

- Once you have successfully generate the executable program, type the command as shown below to run it

```
./hello
```

- If everything is OK, then the message will be printed on screen to show the output of your program.

At this juncture, you should review what you had done in these two simple programming exercises. Make sure that you understand the various steps you used to create and compile the C program to produce an executable file.

5. Displaying message on Sense Hat Module

Instead of displaying message on screen, you can also output message to the Sense Hat module with the RPi-Sense Hat setup. To provide a preview of how the Sense Hat module will be used in subsequent hands-on exercises, you will now code a Python program to display a message on the Sense-Hat module.

- First select the python3 Terminal that you opened earlier on.

- Type the following 3 lines of code in the Terminal, and observe the display on the Sense HAT module.

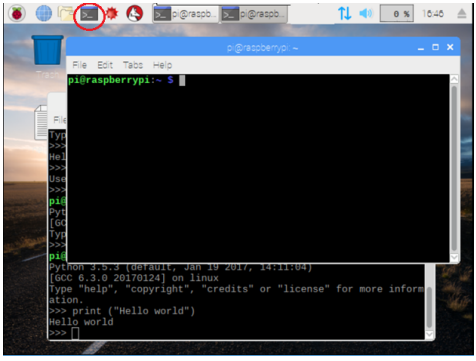
```
>>>from sense_hat import SenseHat
>>>sense = SenseHat()
>>>sense.show_message("Hello NTU")
```

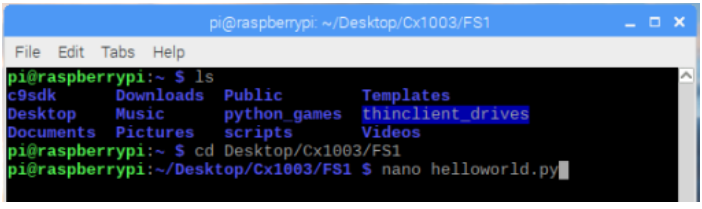
- You can terminate the running interpreter program by pressing the two keys, *Crtl+D*

6. Python in Script Mode

When we write python program for real-world application, we will usually store the python program code in a file, also known as script. We then run the interpreter to execute the script file.

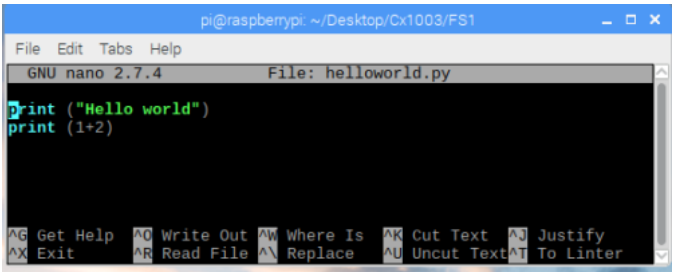
- Open another Terminal, and change directory to the subdirectory of your lab group by typing **cd ~/sc1003/xxx** (replace xxx with your lab group, eg sc1)
- If you can't locate your folder in sc1003 folder, please create your own lab group folder using "mkdir" command (eg.. mkdir sc25)



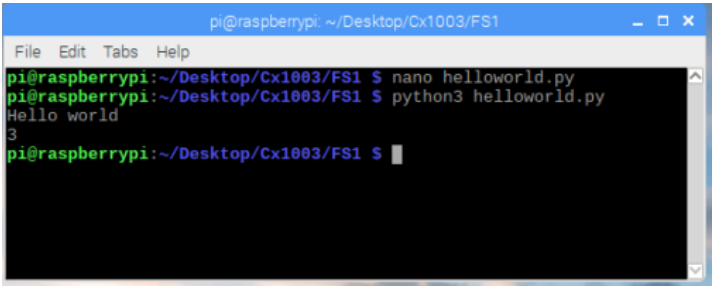


- Use the text editor program, **nano,** create a file calls **helloworld.py**.
- Key in the code as shown below and save the file by pressing the keys **Ctrl and X**. (i.e. *Ctrl* key together with the *X* key), type "**y**" when prompted to 'Save modified buffer?', press [**Enter**] key to confirm save.

- This is the file that contains the program with code written in Python.



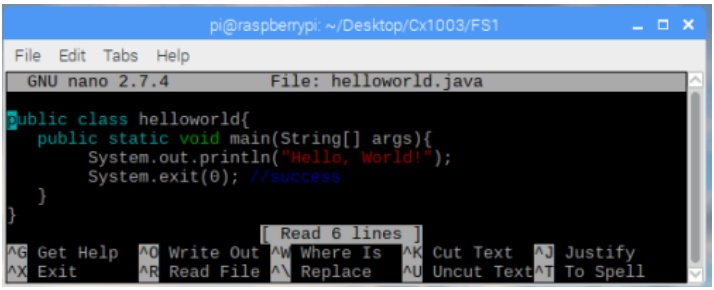
- You can now use the python3 interpreter to run the python program, and you should see the messages shown on the screen.



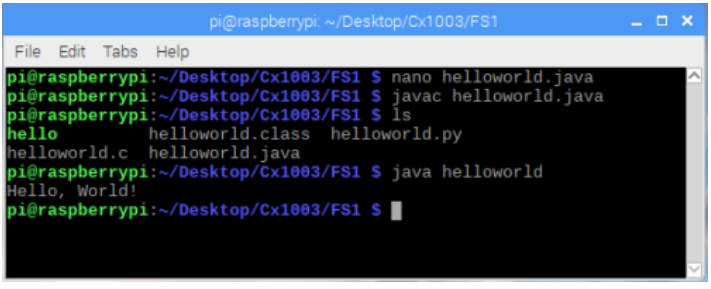
6.1. Basic Java Program

In contrast to Python and C, Java uses both a compiler and an interpreter to run its program. The following exercise shows the basic steps to develop and run a basic Java program.

- Use nano to create a text file *helloworld.java* and key in the code as shown below.



- Next compile the java program by using the ***javac*** compiler as shown below



- A java bytecode file ***helloworld.class*** will be created if there is no error.

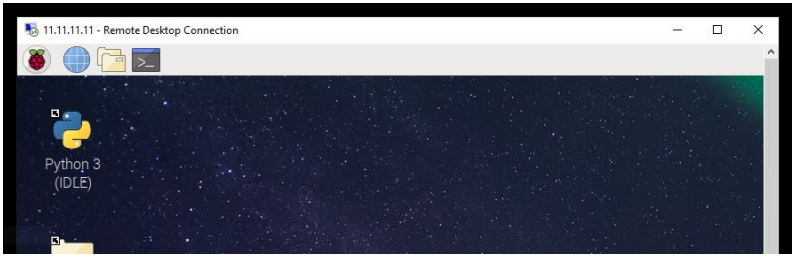
- The bytecode file can then be executed by using the Java interpreter, using the following command

java hellowworld

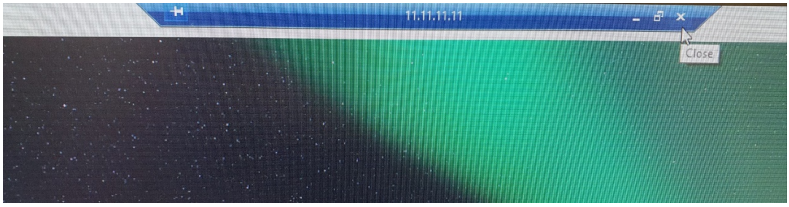
The corresponding result message will then be shown on screen.

6.2. Terminate the remote desktop windows

Before we continue to Lab #1.2, we will temporary terminate the remote desktop window by clicking on the “X”



Or



7. Summary

In the hands-on exercises here, you have learnt how the RPi can be setup as a standalone computer. You have also learnt some of the basic commands used to navigate the RPi system. You have tried writing two basic programs, in Python language and C language to appreciate the difference in interpreted language and compiled language.

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

Laboratory Manual
for
SC1003
Introduction to Computational Thinking and Programming

Practical Exercise #1.2:
Network Access of Raspberry Pi and Cloud9 IDE

Learning Objectives

This practical exercise is to let the students learn how the Raspberry Pi (RPi) can be accessed by a desktop computer (or Notebook) through its Ethernet network connection, and the various software tools that can be used to code programs for the RPi through the remote desktop and execute them on the Raspberry Pi.

Intended Learning Outcomes

At the end of this exercise, you should

- Know the setup required to enable remote computer connection to the Raspberry Pi through its Ethernet interface.
- Be able to access the Raspberry Pi board using software tools such as **PuTTY** and **VNC Viewer**.
- Know how to develop Python programs using the **Cloud9 online IDE**.

Equipment and accessories required

iii) Raspberry Pi 3 Model B (RPi3) board with Sense HAT add-on display module/board.

iv) A USB power source to power the RPi3 board (E.g. Power Bank, Adaptor or ***USB port of a desktop computer***).

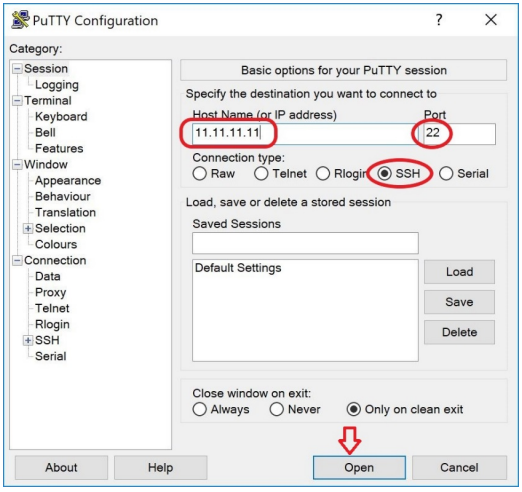
v) A computer (desktop PC or notebook) with Ethernet port and cable for remote access of RPi3. Software (open source) to be installed on the computer – PuTTY, VNC Viewer and WinSCP

2 Remote Access of RPi using PuTTY

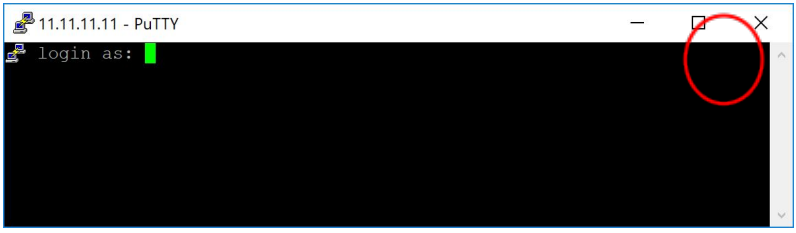
Several programs will be used in this course to access the RPi through the network. To start with, you will use the **PuTTY** program running the Secure Shell (SSH) network protocol.

- Run the **PuTTY** program on the **desktop computer** (which should be already install on the computer)

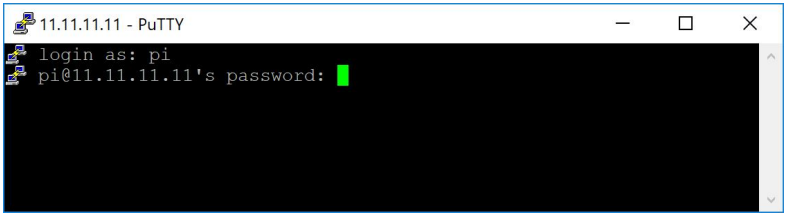
- A pop up window will be shown. Key in the IP address of the RPi, which is pre-set with the value **11.11.11.11**. Make sure that port 22 is selected, and SSH is selected as the connection type.



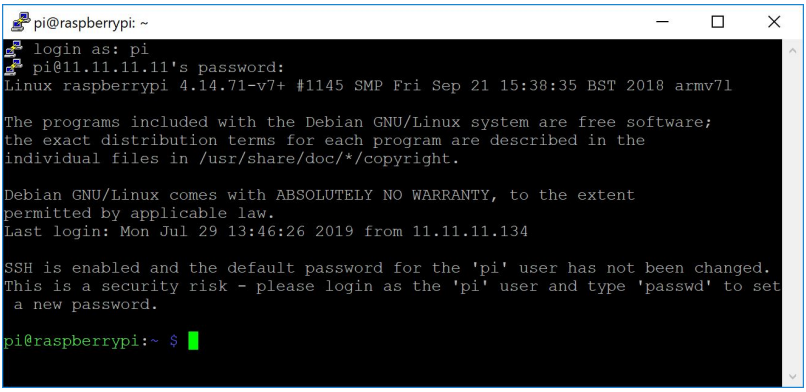
- Click 'Open', and a console interface will appear if the connection is successful. (Otherwise, check the IP address and other settings that you key in.)



- The default login ID is "**pi**" and the password is "**raspberry**". (***Remarks : just type the password normally, you will not see the password been typed or shown***.)



- If the remote login is successful, the console interface display in the PuTTY program will be as shown below, which is essentially identical to the Terminal interface that you used during Practical Exercise #1.



As such, you can now remotely operate the RPi as if you are running the Terminal program directly on the RPi. All the commands that you had used before can be similarly executed through the PuTTY console.

Things to try:

- Write the Python code to print the 'Hello world' message.
- Open another PuTTY console terminal and code the program to display a message on the SenseHat module (refers to manual of Lab Ex #1 for detail).

3 Remote Access of RPi using VNC Viewer

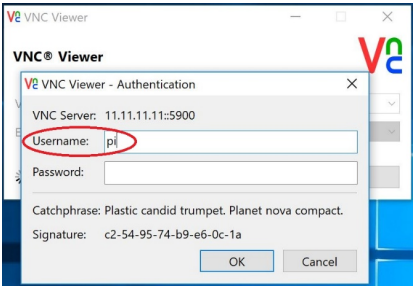
There may be time that you will need to access the RPi through its GUI, such as to launch a web browser to access the internet. For such situation, you can also remotely access the RPI's GUI through the network, using the VNC Viewer program.

3.1 We will first launch the GUI through the VNC Viewer

- Run the VNC Viewer program on the computer
  (It should be already installed).

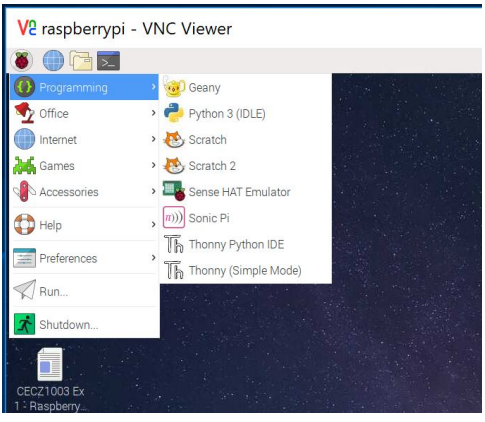
- Key in the IP address of the RPi (which is …). If this is the first time the RPi is accessed by the VNC Viewer program, an Authenticating pop up dialog will appear

to request for the User's ID and password (which are ..)

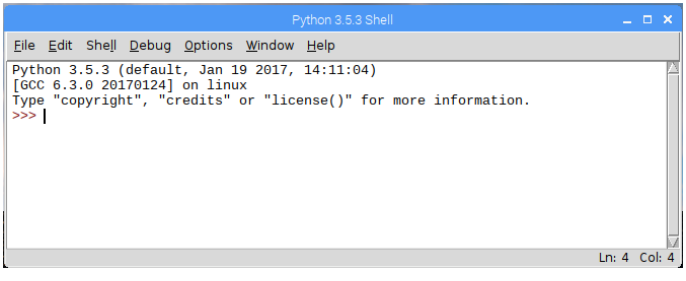


- The GUI of the RPi will then appear in the window, which you can then use to access and operate the RPi as in Practical Exercise #1.

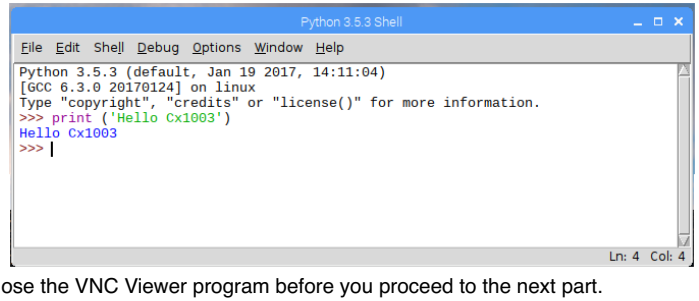
3.2 You will now try using a simple Python IDE (Integrated Development Environment) software with build-in Python interpreter for Python programs development. This is the IDLE program (use the Python 3 version) available under the Application menu.



- It looks the same to the Python3 interpreter running in a terminal, and obviously will behave exactly the same way.



- Try the Hello message code like as shown below. Note that the interface is now displayed in colour to provide better interface to the user.



- Close the VNC Viewer program before you proceed to the next part.

4 Cloud9 IDE

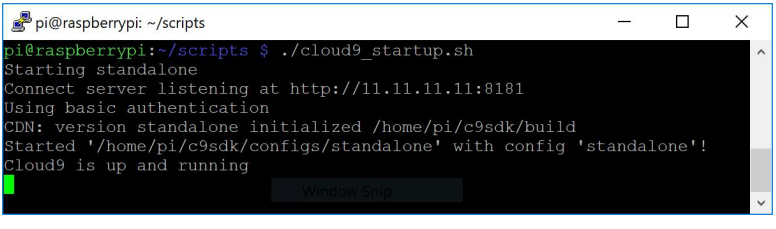
As can be observed, the features provided by the IDLE Python IDE is rather rudimentary, and will not be user friendly once your program grows and becomes more complex. In this course, we will be using a more sophisticated IDE, the **Cloud9 IDE** for you to practice developing the various programming exercises in subsequent practical sessions.

Cloud9 is a 'cloud-based' IDE that lets you write, run, and debug your code with just a web browser. This means that Cloud9 is a web server program that runs remotely on another computer (somewhere behind a 'cloud'), and allows the user to remotely access it using web browser through a network connection. In our case, you will run the Cloud9 Web server on the RPi, and then access it using a web browser on the desktop.

4.1 First you need to **start the Cloud9 web server program on the RPi**. This is to be done by running a (shell script) file that contain the instructions to launch the Cloud9 server program.

- Start a **PuTTY** terminal program on the **desktop computer**, login and change to the scripts directory as shown below. You can use the "ls" command to list the folder content.

- Execute the cloud9_startup.sh shell script file as shown below, and if everything is in order, a message will indicate that the Cloud9 server program is up and running, waiting for user to connect over the network.

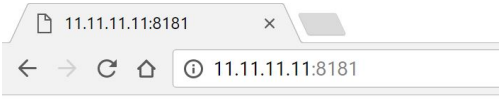


- Note that the program indicates that it is "listening at http://**11.11.11.11:8181**", in which the ***8181*** is the port number it is listening to.

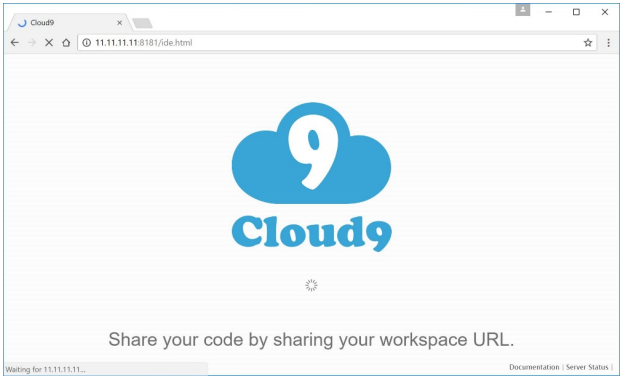
- We will leave this server program running on the RPi, and now try to access it through our desktop computer's browser.

4.2 We will now run a Web browser program on the **desktop computer**, such as the Firefox (or IE, or Chrome).

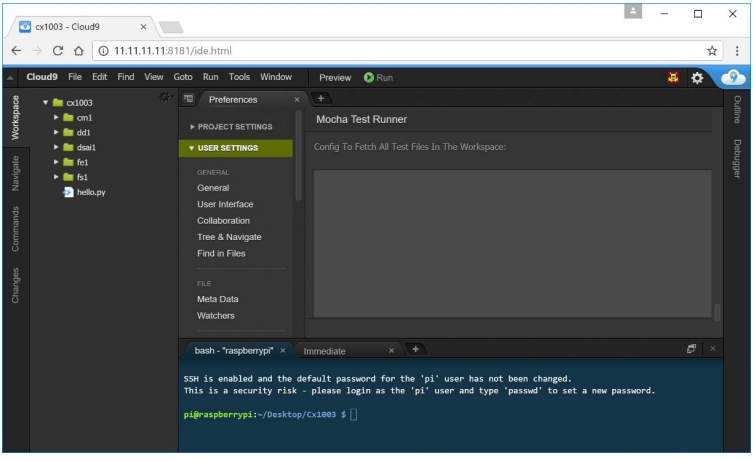
- Open a web browser on the desktop computer, key in the IP address and port number indicated by the server program



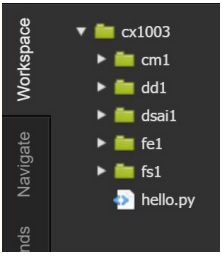
- The Cloud9 screen will appear briefly.



- And eventually the cloud9 IDE interface is launched

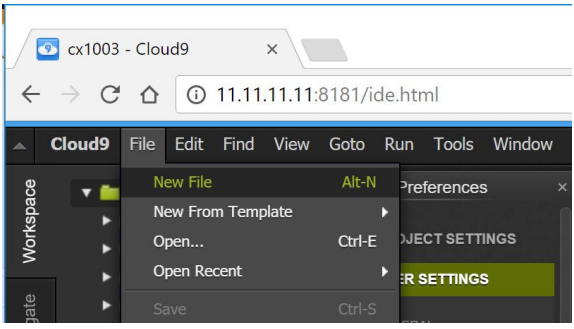


4.3 On the left side of the IDE is the workspace, where you will store the program files you are going to create. The workspace has already been created with various sub-folders corresponding to the different lab groups that you belong to. If you cannot locate your group folder, please create your own group folder. You should store your program files in the appropriate folder as the Rpi board is shared with your classmates.

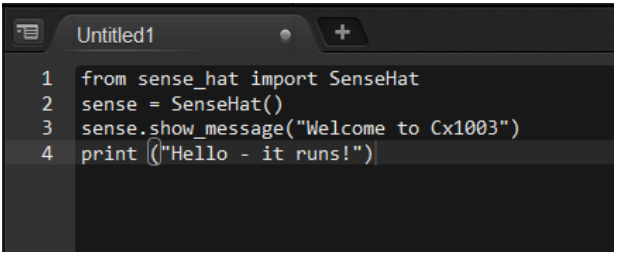


The top right window pane is for you to create your program code.

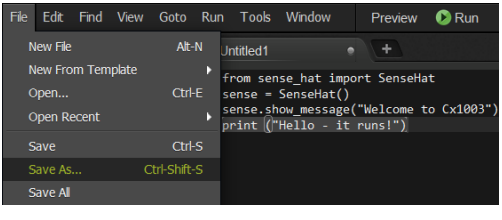
- Click on File→New File option to open a new file for coding

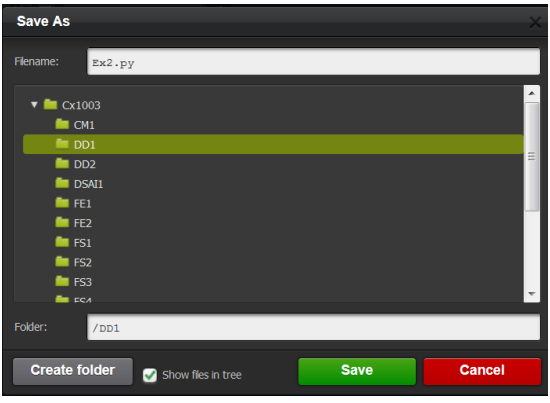


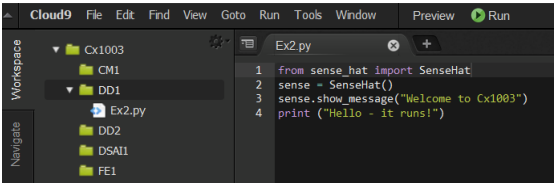
- Type the code such as the following



- Save the file as "**Ex2.py**" in your folder (e.g. DD1 folder here)





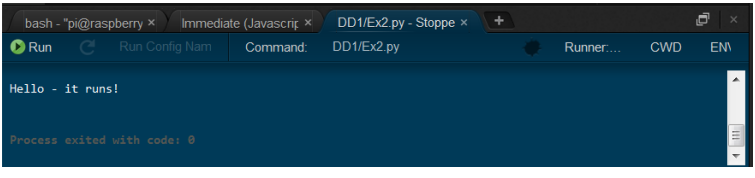


Notice that the IDE now highlights the code in different colours once it knows that this is a Python program. (Q. How does it know that?)

Next you can run the program by clicking the Run icon, 

on the top panel.

- If everything is in order, you should see the message displays on the SenseHat module, as well as the message in the output panel at the bottom of the IDE.



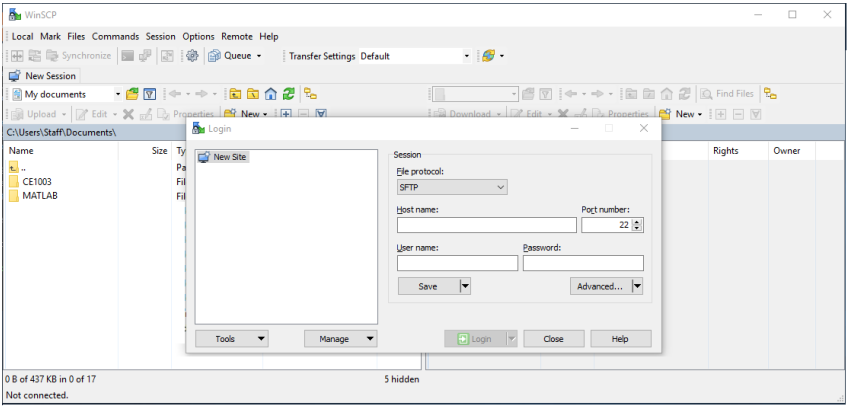
Q. Does the program execution exhibit certain behaviour expected of an interpreted language?

5. Back-up your program file

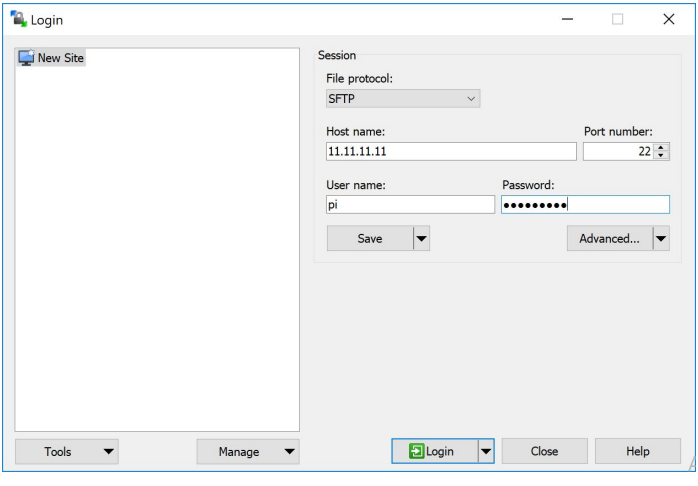
At the end of each practical session, you may want to copy your program files on the RPi to your own USB thumb drive. ***There are several ways to do this***. For example, you can use the WinSCP program as describe here.

- Open the WinSCP program on the desktop computer.

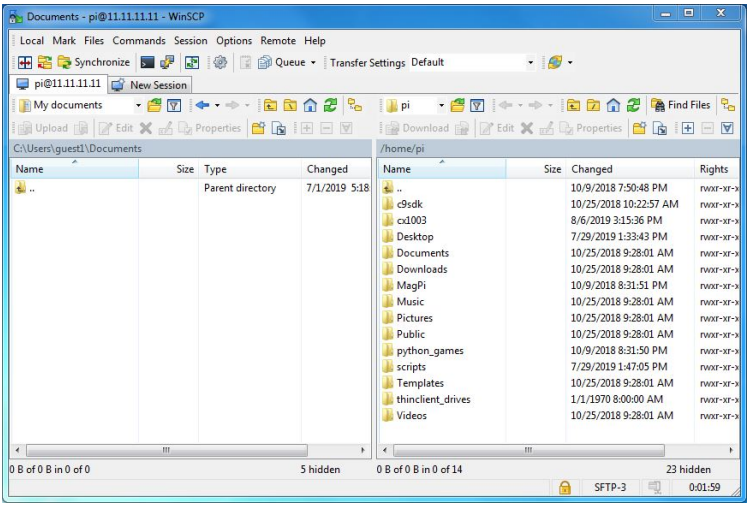
- The program will run and prompt for the detail of the RPi board that you want it to connect to.



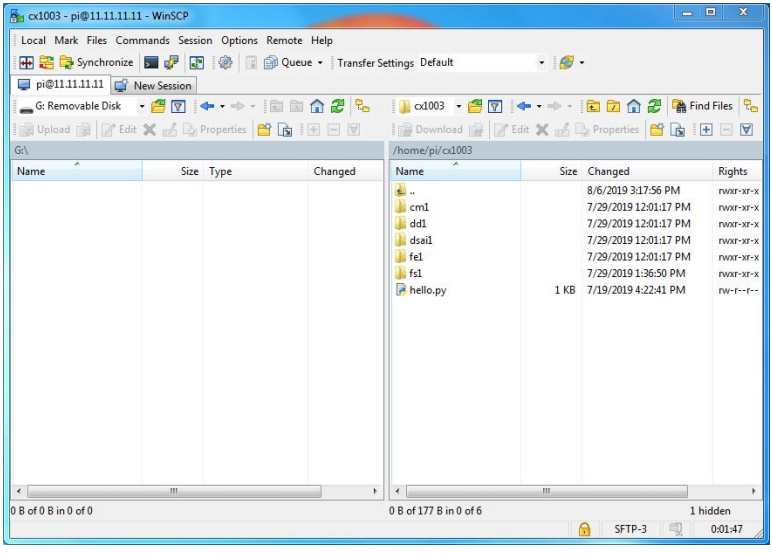
- Key in the information requested



- Once the connection is established, the WinSCP program will display the default directories of both the desktop computer (on the left window pane) and RPi (on the right window pane).



- You can plug in your USB thumb drive to the desktop computer, and then select it for the left window pane. Similarly select the appropriate directory of the RPi on the right window pane.



- You can then copy your file on the RPi to your USB thumb drive by dragging it from the right window pane to the left window pane, and vice versa.

(Alternatively, you can copy your file on the RPi to

the desktop computer, and then email it to yourself.)

Lastly, issue the following commands in the PuTTY Terminal to properly shut down the RPi computer.

- Terminate the Cloud9 IDE – select the PuTTY Terminal that you started the Cloud9 server (you would have observed several messages in the Terminal, correspond to what you had performed earlier in your browser Cloud9 IDE), press **Crtl-C** to stop the server program

If you would like to work on the Appendix below, please do not shut down your RPi, only execute the following shut down procedure after you completed the Appendix.

- Shutdown the RPi by issuing the command: `sudo shutdown –h now`

- **!! Please wait for about 10 seconds for the Rpi to complete the shutdown process before power off the Rpi power (removing usb cables from PC), otherwise system files corruption may occur.**

6. Summary

- In this practical exercise, you had seen how the RPi can be setup in a headless mode. You then learnt how it can be remotely accessed through the network using software programs such as PuTTY and VNC Viewer. You had also learnt how to use the Cloud9 IDE to develop simple Python program, and how to save your work to a USB thumb drive using the WinSCP program.

Appendix - Optional

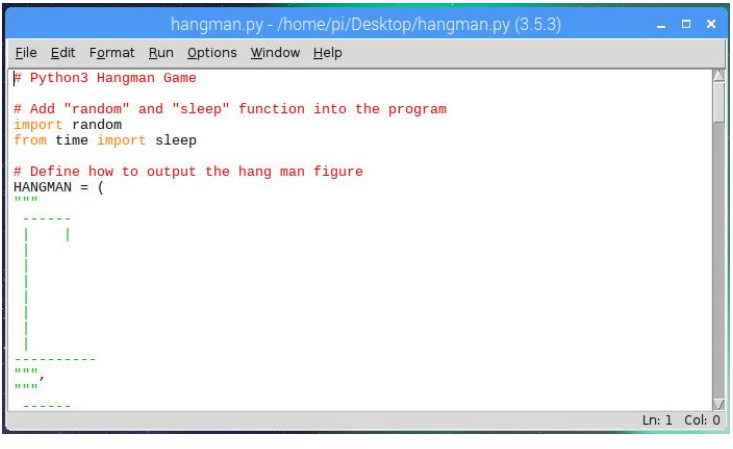
Basic Python game – Have fun !!

It's time to play some game and take a look on how a Python program is coded

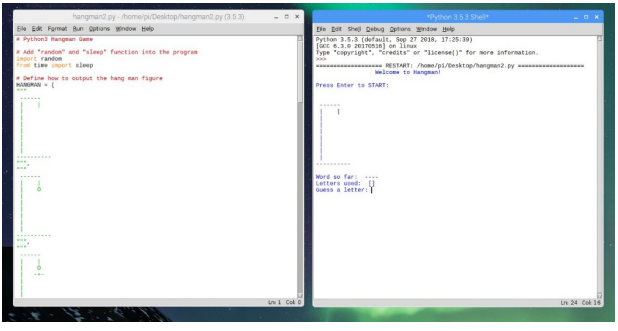
- **copy** the hangman.py program to the Rpi Desktop using the following command in terminal windows,:

cp /usr/local/hangman.py ~/Desktop/hangman.py

- On your Rpi Desktop, double click the "hangman.py" file to open it with Python 3 IDLE editor. Tries to take a look at the codes, it is not difficult to understand what the codes are trying to do.

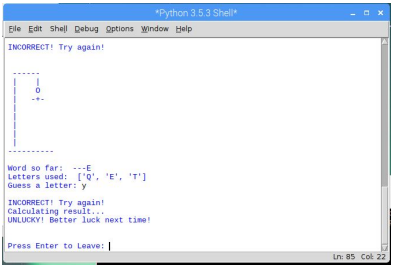


- To execute the code, select “Run” “Run Module F5” from the editor menu bar, this will invoke a new python shell window where you can interact with the game. (Remarks : you can use the <F5> function key to execute the code as well)

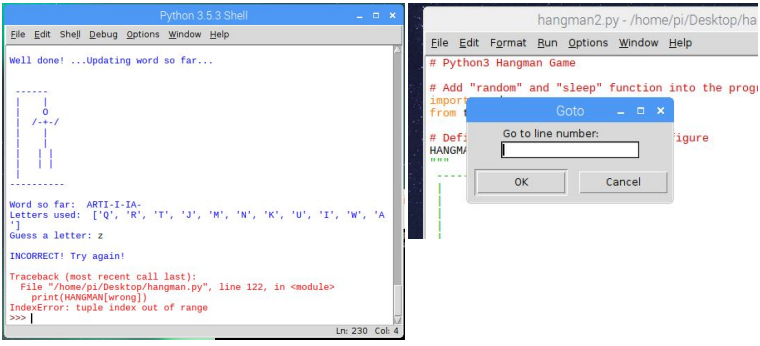


- Tries to play the game a few times, you will find that there are some errors in the codes :

  - The program will end too early before the whole hang man process



  - The program will end with program error message if you complete the hang man process. (Remark : You can use “Edit” “Go to Line” on the python editor to see the line of code that give the error)



- Take a look at the code, and try to modify it to eliminate the error and make the game more functional, a possible end result is as shown below

  Hints :

  - Line 106 : WORDS = ("CODE", "ARTIFICIAL") *** you can add more words

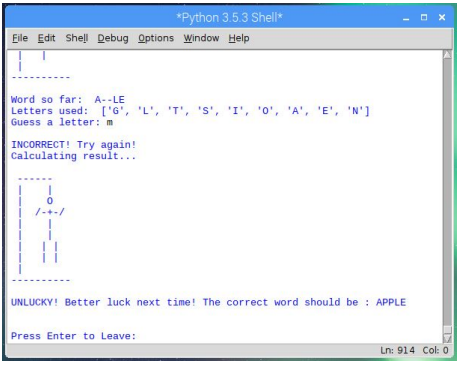
  - Line 110 : MAX_WRONG = len(word) – 1

  - Line 122 : print(HANGMAN[wrong])

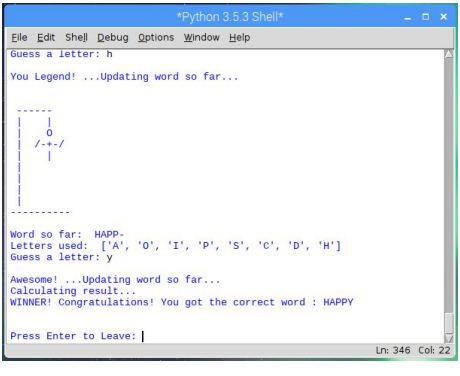
    You may need to add a separate counter for the hangman figure instead of using the “wrong” counter

  - You may need to insert a few more lines of code

Possible game output - *losing*



Possible game out - *winning*



- No worry if you are unable to modify the code, this is just the start of your programming journey, just enjoy and have fun. Very soon you will be able to perfect the code with ease.

- Please delete the hangman.py program on the desktop before shutdown

- Shutdown the RPi by issuing the command in Rpi terminal:

```
sudo shutdown -h now
```