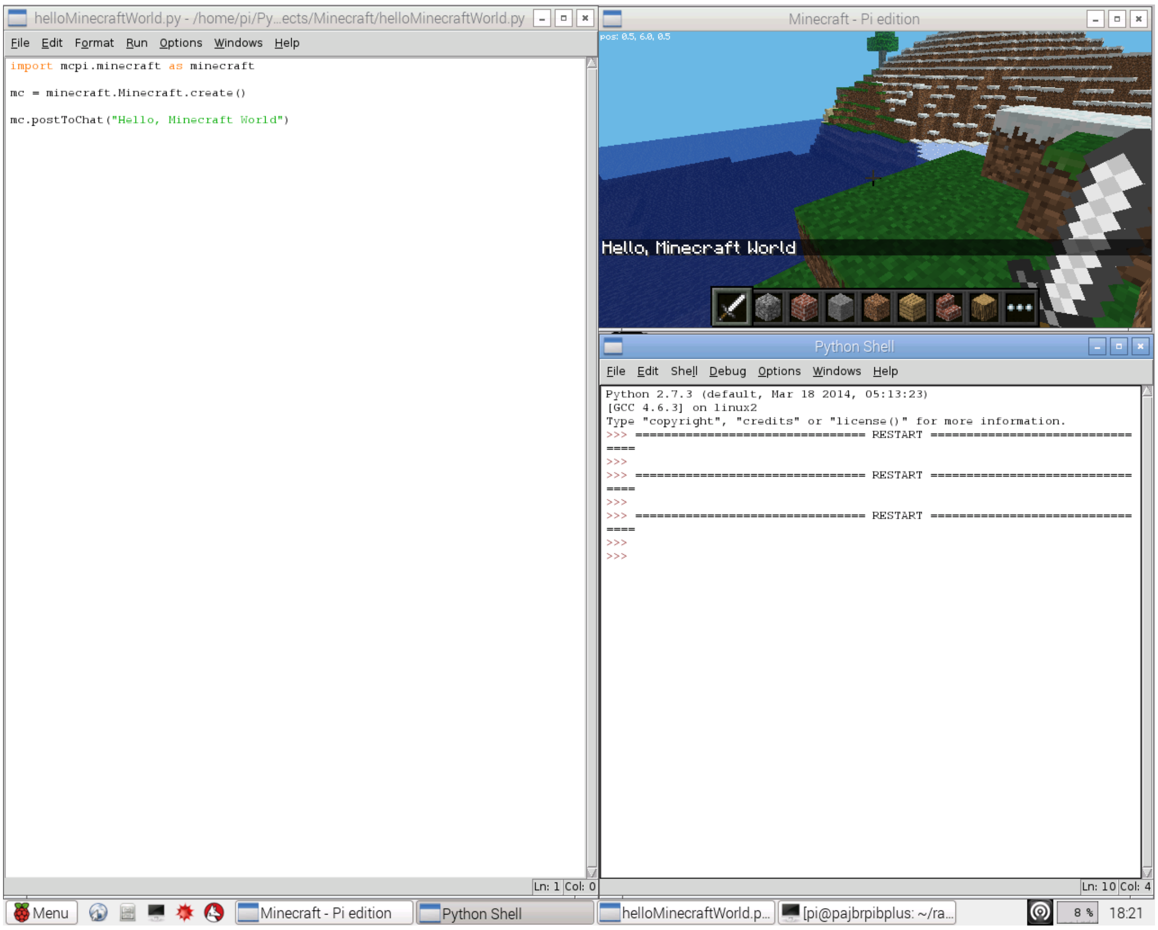
Programming Minecraft Pi with Python

Sheet Two – Hello, Minecraft World

1. Set your screen up by following the “Setting Up” sheet
2. Use File->Save As in the programming window to save a file called “helloMinecraftWorld.py” (and make sure you regularly resave your file as you input any program, particularly when you get on to more complex examples)
3. In the programming window
    a. Import the library which allows us to connect to the Minecraft Pi World
    ```
    import mcpi.minecraft as minecraft
    ```
    b. Create a connection to the Minecraft Pi World
    ```
    mc = minecraft.Minecraft.create()
    ```
    c. Send a message to the player in the Minecraft Pi World
    ```
    mc.postToChat(“Hello, Minecraft World”)
    ```
4. Run the program using Run->Run Module or clicking F5 in the programming window
5. You should see the following


helloMinecraftWorld.py - /home/pi/Py...ects/Minecraft/helloMinecraftWorld.py
File Edit Format Run Options Windows Help
Ln: 1 Col: 0
Minecraft - Pi edition
pos: 0.5, 6.0, 0.5
Python Shell
File Edit Shell Debug Options Windows Help
Ln: 10 Col: 4
Menu
Minecraft - Pi edition
Python Shell
helloMinecraftWorld.p...
[pi@pajbrpibplus: ~/ra...
8 %
18:21

```
import mcpi.minecraft as minecraft

mc = minecraft.Minecraft.create()

mc.postToChat("Hello, Minecraft World")
```

Hello, Minecraft World

```
Python 2.7.3 (default, Mar 18 2014, 05:13:23)
[GCC 4.6.3] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ============================
====
>>>
>>> ================================ RESTART ============================
====
>>>
>>> ================================ RESTART ============================
====
>>>
>>>
```

Notes

Importing Libraries

We can make use of Python code that other people have written, saving us from having to write everything from scratch. In this example we are using a pre-existing library to connect to the Minecraft Pi program.

Before we can make use of a library we need to import it. Every library has a unique name. The library for connecting to the Minecraft Pi program is called `mcpi.minecraft` and we could import it like this. Whenever we wanted to make use of it we would need to type `mcpi.minecraft` in full again.

```
import mcpi.minecraft
```

Alternatively, we can give it a different name, in this case simply `minecraft`, by adding an `as` clause. As you will see, now we can use this library by typing `minecraft` instead of `mcpi.minecraft.`

```
import mcpi.minecraft as minecraft
```

Objects and Methods/Functions [check Python terminology]

Libraries expose their functionality (technically called an Application Programming Interface or API) by a set of objects and functions. The `mcpi.minecraft` library contains an object called `Minecraft` which has a function called `create()` that returns an object we can use to interact with the Minecraft Pi World.

Variables

A variable is a "named slot" that we can hold a value in. In this case, we are defining a variable called `mc` which will hold the connection that we have created to allow us to interact with the Minecraft Pi World

```
mc = minecraft.Minecraft.create()
```

Method/Functions [check Python terminology] Revisited

Now that we have a variable which is connected to the Minecraft Pi World, we can call the methods on the variable in order to interact with the Minecraft Pi World. One such method (defined and implemented by the author of the `mcpi.minecraft` library) allows us to post a message to the screen.

Methods can take one or more arguments which affect their behaviour. The `postToChat()` method takes a single argument – a string containing the text to be displayed. Without arguments you would need a different method for each and every string you might want to show!

```
mc.postToChat("Hello, Minecraft World")
```