

# MINECRAFT WORKSHOP

This workshop will guide you through navigating and controlling the Minecraft world directly with code. The techniques presented here can be combined with other workshops (control things by hitting Minecraft blocks!)

## WHY BUILD, WHEN YOU CAN CODE?

If your Raspberry Pi is at the command line and not the GUI, login then type `startx` and press Enter. Click **Menu > Games > Minecraft Pi** to start the game.

## HAVE A WANDER!

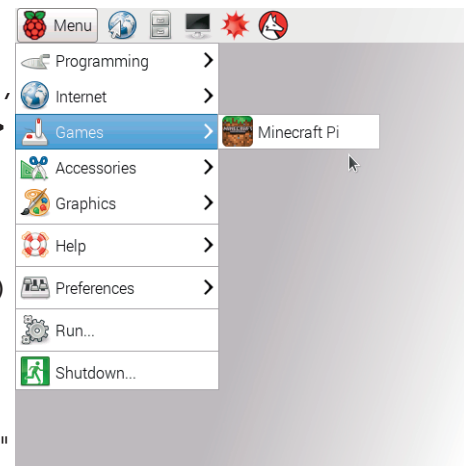
Click **Start Game**, then click **Create New** (or choose an existing one) to enter a world:

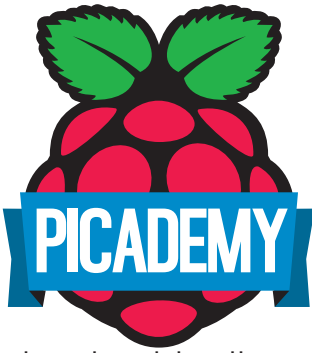
- The mouse changes where you look
- Holding the left mouse button destroys blocks
- The right button hits blocks (with the sword) or "places" blocks (with anything else selected)
- W, S, A, D move you forward, backward, left and right respectively
- E opens the inventory where you can select blocks and tools
- Space is jump; double tapping Space toggles flying
- While flying Space goes up, Shift goes down
- Esc takes you back to the menu
- Tab releases the mouse to allow you to control the desktop

## THE API

The Minecraft API works by talking to the "server", which runs underneath the game, allowing you to change the game as it is being played. For example, you can:

- Post messages to the player
- Get and change the player's position
- Get and change the type of a block
- Detect when the player hits a block with their sword

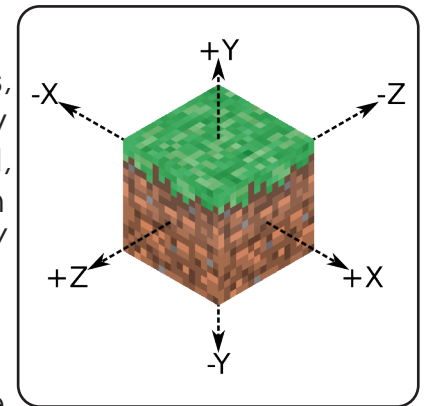




# MINECRAFT WORKSHOP

## A WORLD OF BLOCKS

Minecraft is a world of cubes or blocks, all with a relative size of  $1\text{m}^3$ . Every block has a position in the world, determined by its x, y, z coordinates. The X and Z axes run horizontally (think of them as describing the ground), while the Y axis is vertical.



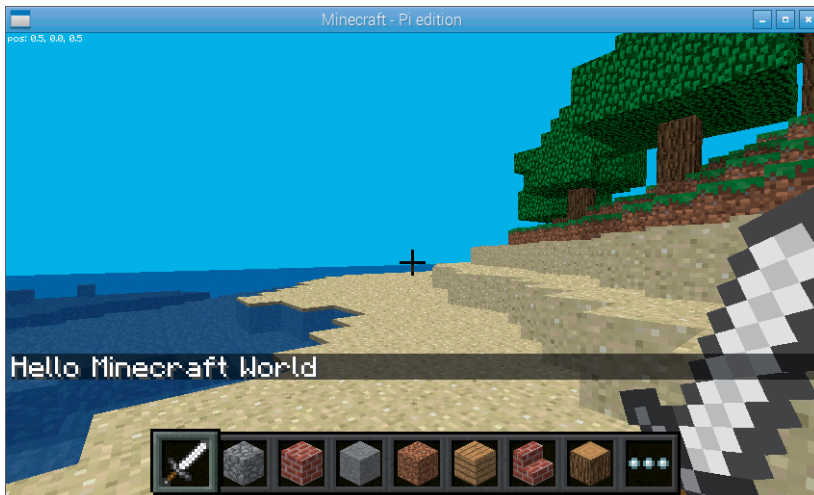
## HELLO MINECRAFT WORLD!

In this exercise, you will display "Hello Minecraft World" on the game's chat console. This is primarily to check that everything is working.

Open Python 3 by clicking **Menu > Programming > Python 3**. Use **File > New Window** to create a new program and save it as hello\_minecraft.py. Type the following code into your new program:

```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()
mc.postToChat("Hello Minecraft World")
```

The first line imports the minecraft module from the Minecraft: Pi Edition library into your program. The second line creates a connection to the Minecraft server. Finally, the third line uses the connection to send the player a message using postToChat.



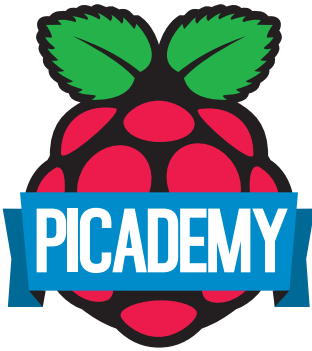
Run the program by selecting **Run > Run Module** in Python. Switch back to Minecraft to see the results of your hard work!

## EXTRAS

Can you make the script display "Hello Minecraft World" 10 times?



Unless otherwise stated, all content is under a Creative Commons Attribution-ShareAlike 4.0 International License. All resources can be found at [https://github.com/waveform80/picademy\\_worksheets/](https://github.com/waveform80/picademy_worksheets/)



# MINECRAFT WORKSHOP

## DON'T LOOK DOWN!

Move Steve (the player) up in the air and let him fall to the ground, by adding 50 to his position on the Y-axis. Add insult to injury by posting "Don't look down!" to the chat window as he falls.

Use **File > Save As** in the menu to save your script as `dont_look_down.py`, then edit it to use `getPos` to obtain Steve's position and store it in the `pos` variable. Finally, use `postToChat` to display the position in the chat console:

```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()
pos = mc.player.getPos()
mc.postToChat(pos.x)
mc.postToChat(pos.y)
mc.postToChat(pos.z)
```

Run the program by selecting **Run > Run Module** to see Steve's position. You can now add some lines to your program to change Steve's position to be 50 blocks higher using `setPos` to add 50 to his Y position:

```
mc.player.setPos(pos.x, pos.y + 50, pos.z)
mc.postToChat("Don't look down!")
```

Run the program by clicking **Run > Run Module**.



## EXTRAS

Can you change the script so Steve is only flung into the air if his Y position is greater than 10?

## THIRD PERSON

From the Escape menu, you can change your view of the Minecraft world. Click on the icon at the top left and you'll switch to "third person" view. Click the icon again to switch back to "first person" view.



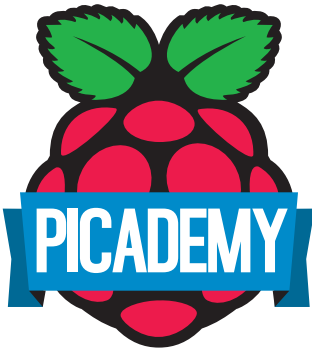
First person view



Third person view



Unless otherwise stated, all content is under a Creative Commons Attribution-ShareAlike 4.0 International License. All resources can be found at [https://github.com/waveform80/picademy\\_worksheets/](https://github.com/waveform80/picademy_worksheets/)



# MINECRAFT WORKSHOP

## THE DUPLICATOR

Steve needs blocks! When your next program runs, it should duplicate the block Steve is standing on, and put it on his head.

Use **File > Save As** to save your program as duplicator.py. Now get rid of all but the first three lines so your program simply connects to Minecraft. This time instead of using getPos, we'll use getTilePos to obtain Steve's position in block coordinates (these are always whole numbers):

```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()
pos = mc.player.getTilePos()
```

All blocks in Minecraft have a number (see the Blocks side panel). Use getBlock to find out what type of block is below Steve, and store the number in the variable below. Finally, post that number to the chat window:

```
below = mc.getBlock(pos.x, pos.y - 1, pos.z)
mc.postToChat (below)
```



Run your program with **Run > Run Module** a few times while standing on different types of block; you should see different numbers. Next, add to your program so that the block above Steve is the same as the one below using the setBlock call:

```
mc.setBlock(pos.x, pos.y + 2, pos.z, below)
```

Run the program and remember to look up to see if the block above Steve has changed!

### EXTRAS

Can you duplicate all the blocks in a 3x1x3 square underneath Steve?

## BLOCKS

Each type of block in Minecraft is identified with a different number. Some of the more common types are listed on the right with their identifying numbers in brackets. Some blocks behave differently too. For example, sand blocks always fall (so if you duplicate one above you it will fall through you!), while stone and grass don't. Water and lava blocks flow down from where you place them.



Grass (2)



Stone (1)



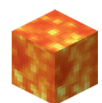
Sand (12)



Diamond (57)



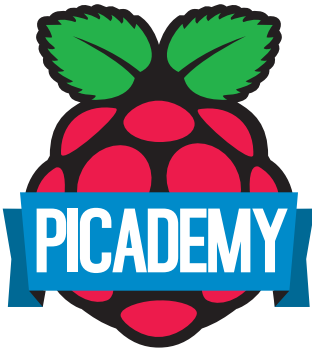
Water (8)



Lava (10)



Unless otherwise stated, all content is under a Creative Commons Attribution-ShareAlike 4.0 International License. All resources can be found at [https://github.com/waveform80/picademy\\_worksheets/](https://github.com/waveform80/picademy_worksheets/)



# MINECRAFT WORKSHOP

## I'M WALKING ON SUNSHINE DIAMOND!

All that walking up and down hills is a pain. Create a diamond bridge which automatically appears under Steve's feet when he walks.

Use **File > New** to create a new program, and **File > Save As** to store it as `walk_on_air.py`. Now, import the minecraft and block modules, and connect to the game as in the previous exercises:

```
from mcpi.minecraft import Minecraft
from mcpi import block
mc = Minecraft.create()
```

### LOOPS

Python should automatically indent the code under the loop when you press Enter after the colon (:), but if not simply press Tab before typing in the code for the loop.

```
while True:
    pos = mc.player.getTilePos()
    mc.setBlock(pos.x, pos.y - 1, pos.z, block.DIAMOND_BLOCK)
```

The block module is useful when creating blocks as you can use real block names, like `AIR`, rather than having to remember that air is block type 0.

Create a while loop which will continue forever. Indented under the while loop, constantly ask Minecraft for Steve's position and set the block under his feet to a DIAMOND\_BLOCK:

Run the program with **Run > Run Module** and walk off a cliff! To stop the script, select **Shell > Restart Shell** from the Python Shell window.



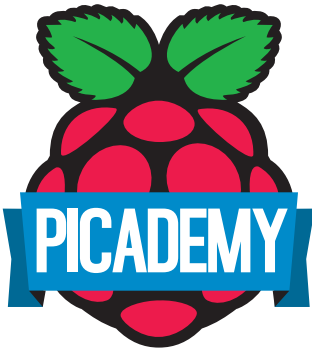
### EXTRAS

Can you alter the script to only create a bridge under Steve if the block below him is air?



Unless otherwise stated, all content is under a Creative Commons Attribution-ShareAlike 4.0 International License. All resources can be found at [https://github.com/waveform80/picademy\\_worksheets/](https://github.com/waveform80/picademy_worksheets/)





# MINECRAFT WORKSHOP

## NEVER MIND THE PEN...

program and

module, and connect to the game as in previous exercises:

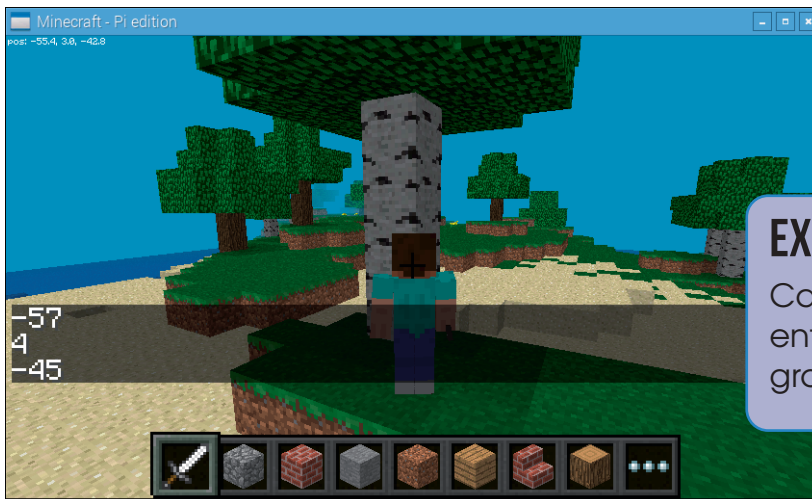
You can ask the Minecraft world whether Steve has hit any blocks with his sword (remember you have to right-click to hit blocks; if you left-click you'll destroy them!). Use **File > New** to create a new **File > Save As** to store it as hit\_blocks.py. Now import the minecraft

```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()
```

Once again, create a while loop which continues forever. Indented under the while loop, constantly ask the server for events by calling pollBlockHits. This returns a list of all the events that occurred since the last call to pollBlockHits, so we need another loop to deal with this list.

This time we use a for loop which doesn't continue forever; it just loops over the list of events, and posts the coordinates of the block that was hit to the chat console:

```
while True:
    for event in mc.events.pollBlockHits():
        mc.postToChat(event.pos.x)
        mc.postToChat(event.pos.y)
        mc.postToChat(event.pos.z)
```



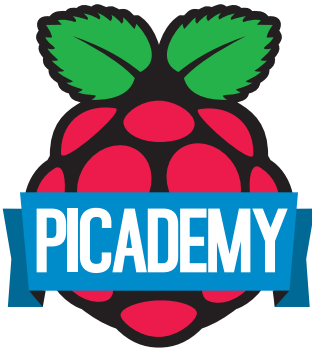
You can also use event.face to determine which side of the block was hit.

## EXTRAS

Can you alter the script to post different messages when Steve hits sand or grass?



Unless otherwise stated, all content is under a Creative Commons Attribution-ShareAlike 4.0 International License. All resources can be found at [https://github.com/waveform80/picademy\\_worksheets/](https://github.com/waveform80/picademy_worksheets/)



# MINECRAFT WORKSHOP

## STRETCH GOALS!

If you've got the time, try stretching your skills with some of the extra exercises below...

## RAINBOW BRIDGE

You can get coloured blocks by using WOOL. Change your diamond bridge program to use coloured WOOL by passing a number 0-15 (0="white", 1="orange", 2="magenta", etc.) after the block id:

```
mc.setBlock(x, y, z, block.WOOL.id, wool_color)
```

How about random wool colours using Python's random number module and randint function to return a number between 0 and 15:

```
import random
number = random.randint(0, 15)
```

## TRAFFIC LIGHTS

Use your new wool blocks to create a working traffic light (turn your traffic lights off with black wool):

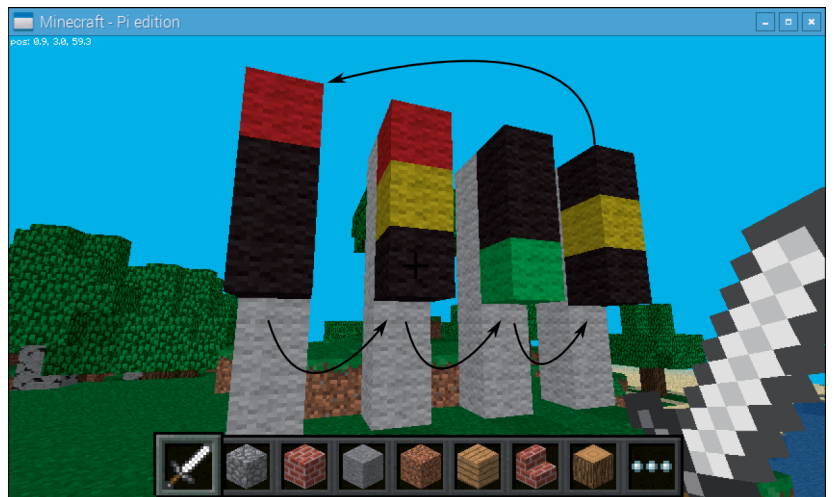
```
# create red wool
mc.setBlock(x, y, z, block.WOOL.id, 14)
# red = 14
# yellow = 4
# green = 13
# black = 15
```

You can put delays in your program using Python's time module and the sleep function:

```
import time
# sleep for 1 second
time.sleep(1)
```

### FIND OUT MORE

- [pi.minecraft.net](http://pi.minecraft.net)
- [www.stuffaboutcode.com](http://www.stuffaboutcode.com)
- [blog.whaleygeek.co.uk](http://blog.whaleygeek.co.uk)
- [arghbox.wordpress.com](http://arghbox.wordpress.com)
- [mcpipy.wordpress.com](http://mcpipy.wordpress.com)



Unless otherwise stated, all content is under a Creative Commons Attribution-ShareAlike 4.0 International License. All resources can be found at [https://github.com/waveform80/picademy\\_worksheets/](https://github.com/waveform80/picademy_worksheets/)