



*“Why build when you can code”*

## Install the game

Open up a LX Terminal Window:

```
cd ~
```

Download from the Internet:

```
wget https://s3.amazonaws.com/assets.minecraft.net/pi/minecraft-pi-0.1.1.tar.gz
```

Copy from USB Stick:

```
sudo mkdir /media/usbstick  
sudo mount -t vfat -o uid=pi,gid=pi /dev/sda1 /media/usbstick/  
cp /media/usbstick/minecraft-pi-0.1.1.tar.gz /home/pi  
sudo umount /media/usbstick
```

Install:

```
tar -zxvf minecraft-pi-0.1.1.tar.gz
```

## Run the game

```
cd ~/mcpi
```

```
./minecraft-pi
```

## Have a wander about!

The mouse changes where you look.

Holding left button destroys blocks.

Right button ‘places’ blocks

W, S, A, D move you forward, backward, left and right.

E opens the inventory

ESC takes you back and to the Menu

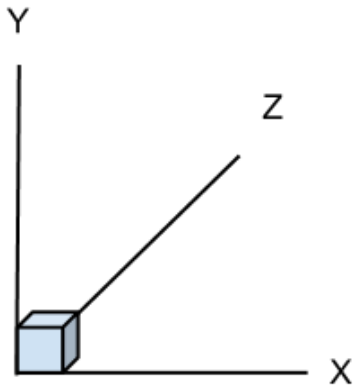
## The API

The API works by changing the ‘server’, which runs underneath the game, allowing you to interact with these blocks and the player, such as:

- Post messages to the player
- Get and Change (or set) the player’s position
- Get the type of a block
- Change a block

## A world of blocks

Minecraft is a world of cubes or blocks, all with a relative size of 1m x 1m x 1m, and every block has a position in the world of x, y, z; x and z being the horizontal positions and y being the vertical.



## Hello Minecraft World

Open IDLE (not IDLE 3), use File, New Window to create a new program and save it as myprogram.py in the ~/mcpi/api/python/mcpi directory.

You are going to need 3 modules, the minecraft.py and block.py modules from the api library and the standard time module so you can introduce delays into your program.

```
import minecraft
import block
import time
```

Use the Minecraft class in the python library to create a connection to the game's server, this object will also be how we interact with the game and will provide access to all the functions.

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

Using the minecraft object, mc, you can interact with the game and send the player a message. Then using time put in a delay to stop the program running too fast.

```
mc.postToChat("Hello Minecraft World")
time.sleep(5)
```

Run the program by picking 'Run, Run Module' in Idle.

## Falling down

Move Steve (the player) up in the air and let him fall to the ground, by adding the following to your program.

Find out where Steve is using `mc.player.getPos()`

```
playerPos = mc.player.getPos()
```

Change Steve's position 50 blocks up using `mc.player.setPos` and adding 50 to his Y position.

```
mc.player.setPos(playerPos.x, playerPos.y + 50, playerPos.z)
```

Put a message on the screen and wait for Steve to fall to the ground.

```
mc.postToChat("Dont look down")  
time.sleep(5)
```

Run the program by picking 'Run, Run Module' in Idle.

## Changing Blocks

Turn the block which Steve is standing on into Diamond. Find out which block Steve is standing on by using `mc.player.getTilePos()`

```
playerTilePos = mc.player.getTilePos()
```

Change the block below Steve using `mc.setBlock()`

```
mc.setBlock(playerTilePos.x, playerTilePos.y - 1, playerTilePos.z,  
block.DIAMOND_BLOCK.id)
```

Run the program by picking 'Run, Run Module' in Idle.

## Walking on Air

Change our program so that it loops infinitely changing the block below Steve.

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

## Rainbow blocks

You can get coloured blocks by using WOOL, lets change our program to use random coloured WOOL by passing a number 0-15 (0="white", 1="orange", 2="magenta", etc) after the block id. To create random numbers, we will need to import python's random module at the top of the program..

```
import random
```

Change the last setBlock command to create WOOL instead of DIAMOND\_BLOCK and pass a random number between 0 and 15 for the wool colour.

```
woolColour = random.randrange(0,15)
mc.setBlock(playerTilePos.x, playerTilePos.y - 1,
playerTilePos.z, block.WOOL.id, woolColour)
```

## Summary of some useful API functions

- postToChat(message) - communicate with the player(s) in the game
- getBlock(x, y, z) - get a block type for a specific position
- setBlock(x, y, z, blockType, blockData) - set (change) a block to a specific blockType
- setBlocks(x1, y1, z1, x2, y2, z2, blockType, blockData) - set lots of blocks all at the same time by providing 2 sets of co-ordinates (x, y, z) and fill the gap between with a blockType
- player.getPos() - get the precise position of a player
- player.setPos(x, y, z) - set (change) the players position
- player.getTilePos() - get the position of the block where the player current is

## Have a go yourself

Use the functions above to have a go yourself!

## Some Ideas

Build something of your own design (a house, a flower, a badger, anything).

Tip - you might find it easier using

`setBlocks(x1,y1,z1,x2,y2,z2,block.STONE.id)` which allows you to create lots of blocks all at the same time.

Create an animation - make blocks and shapes move.

Tip - to make a block disappear you need to set it to AIR, `setBlock(x,y,z,block.AIR.id)`

Martin O'Hanlon  
@martinohanlon  
[www.stuffaboutcode.com](http://www.stuffaboutcode.com)