

What you've learned

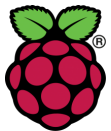
Maths operators Addition, subtraction, multiplication and division are all possible in Python. Using them in Python allows you to change the values of variables or arguments. For example we used addition on lines 9–11 to add to the values of `x`, `y` and `z` and store the result in new variables. We also used addition and subtraction to modify the values of arguments on line 13.

setBlocks() The `setBlocks()` function of the Minecraft Pi API creates a cuboid of blocks between two sets of co-ordinates. These co-ordinates are provided as arguments. In the example we used six variables for these arguments. The seventh argument is the type of block that will be created.

Your Turn!

Here are some suggestions to extend your code and make it do different things. Try anything you think of too!

- Change the block types of the blocks being placed. A popular block type for the inside is lava (value 11) with an surrounding of glass (value 20).
- Make a swimming pool. Change the values on the maths operators used in the `setBlocks()` function on line 13 so that there is no top roof created.



Raspberry Pi and the Raspberry Pi logo are trademarks of the Raspberry Pi Foundation <http://www.raspberrypi.org>. Minecraft is a registered trademark of Mojang.

These resources are copyright Craig Richardson and licensed under a [Creative Commons BY-NC-SA License](#). These resources have been adapted by Code Park.



Warehouse

When you're building in Minecraft you can spend ages on a relatively simple structure, such as the walls of your house. Wouldn't it be nice if you could save time? Well, you can do that with only a few lines of Python.

In this guide you'll learn how to use maths operators in Python to create the basic structure for a house. You'll create a hollow box of block, which you can build upon to create your own ideas with more speed.

You can have loads of fun adapting this code. Come up with your own ideas and try changing the block types or combining it with other bits of code from other exercises.

Code

Import the API and set variables

As usual we import the API and connect to the game. The `block` and `air` variables store block types, which we will use later. Block type 4 is cobblestone.

```
1  from mcpi.minecraft import Minecraft
2  mc = Minecraft.create()
3  block = 4
4  air = 0
```

Set variables

The `x`, `y` and `z` variables represent the bottom left hand corner of a 3D rectangle, known as a cuboid. The variables `x2`, `y2` and `z2` add to the `x`, `y` and `z` variables and store the new values. The `+` operator is used for addition. Likewise `-` can be used for subtraction.

```
5  x = 10
6  y = 11
7  z = 12
8  x2 = x + 10
9  y2 = y + 5
10 z2 = z + 8
```

Create blocks

Similar to `setBlock()`, the `setBlocks()` function creates blocks in a Minecraft game. It creates a cuboid (3D rectangle) between two points. These points have been set using the variables that we created on lines 5–10. We also provide the block variable to say what type of blocks should be created in the cuboid.

```
11 mc.setBlocks(
12     x, y, z,
13     x2, y2, z2,
14     block
15 )
```

Create air blocks

We once again call the `setBlocks` function. This time we use addition and subtraction to create a cuboid made of air within the cuboid we created on lines 11–15. The `air` variable sets the cuboid to air, which effectively hollows the original cuboid, leaving a ceiling, floor and walls that are one block thick.

```
16 mc.setBlocks(
17     x + 1, y + 1, z + 1,
18     x2 - 1, y2 - 1, z2 - 1,
19     air
20 )
```