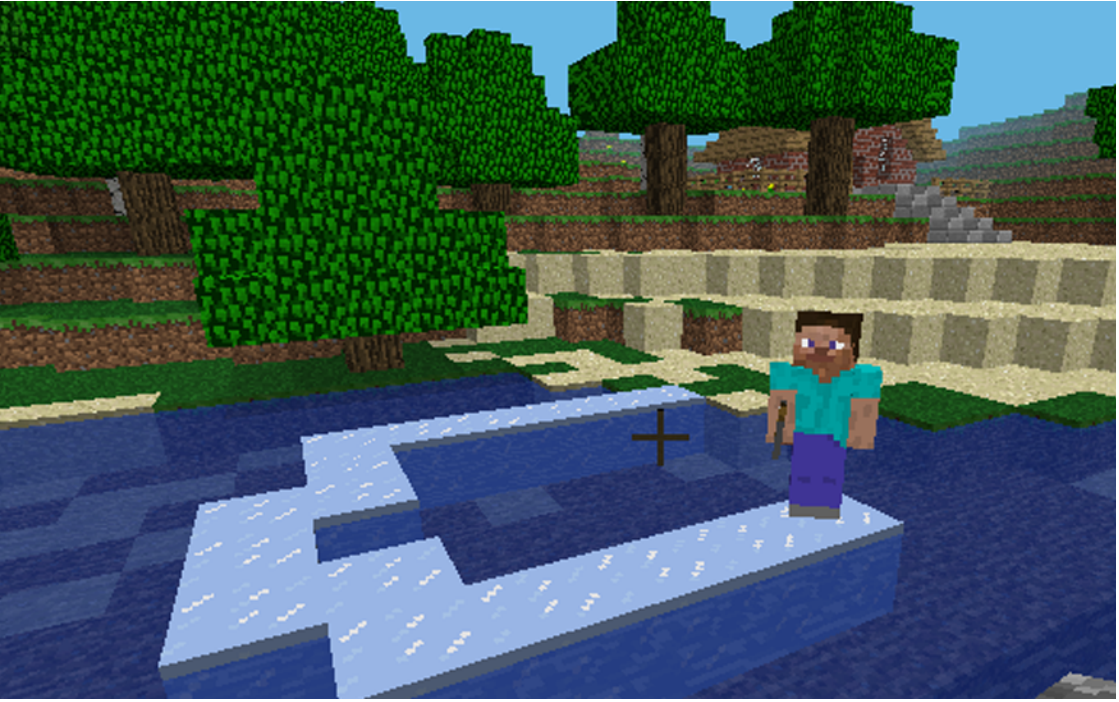


5. Freeze

One of the benefits of programming is that it can automate decision making. This saves us time and allows us to create rules that our programs must follow.

In this guide we'll use Python code to freeze any blocks of water below the player. By using decision making in Python, we can avoid freezing any other blocks than water.

By making small changes to the code you can change the effect of the program quite drastically. For example you could make the program change grass to gold. Try coming up with your own ideas!

Code

Import the API

As usual we import the API and make a connection to the game. We also import time.

Start the loop and get the player's position

As we need to repeat the code constantly we use an infinite `while` loop. We slow down the speed that the loop repeats with `time.sleep()`. On line 6–9 we find out the player's position and store it in the `x`, `y` and `z` variables.

Get the block below the player

To find the type of a block in the Minecraft game we use the `getBlock()` function. It uses co-ordinates and returns the block's type at those co-ordinates. To find the block below the player we subtract 1 from the `y` variable.

Create block variables

On lines 10–12 we create variables to store the block types of water and ice, which we will use later.

Check the block below the player

To check the block below the player we use an `if` statement. An `if` statement only executes some code when something is `true`. In this case it checks whether the block below the player (stored in the `blockBelow` variable) is water. If the block below the player is water it will set the block below the player to ice.

```
1  import mcpi.minecraft as minecraft
2  mc = minecraft.Minecraft.create()
3  import time
```

```
4  while True:
5      time.sleep(0.2)
6      pos = mc.player.getPos()
7      x = pos.x
8      y = pos.y
9      z = pos.z
```

```
9      blockBelow = mc.getBlock(x, y - 1, z)
```

```
10     water = 9
11     ice = 79
```

```
13     if blockBelow == water:
14         mc.setBlock(x, y - 1, z, ice)
```

What you've learned

getBlock() The `getBlock()` function finds the type of a block at certain co-ordinates. The co-ordinates are given as arguments to the function, for example in this program we use the `x`, `y` and `z` variables as these arguments. The function returns the block type at those co-ordinates.

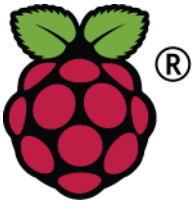
if statement An `if` statement will only run a section of code when a condition is `True`. When the condition is `False`, the section of code will not run. For example in our program the block below the player will only turn to ice if it is water.

Equal to (==) The equal to operator checks whether one value is the same as the other. If they are the same it will evaluate to `True`, if they are not it will evaluate to `False`. In our program we use an equal to operator with an `if` statement to check whether the block below the player is water.

Extensions

Here are some suggestions to extend your code and make it do different things. Even better if you come up with your own ideas.

- Change the block type that is changed. Try changing it to lava (block value 10).
- Check that the block below the player is not air (block value 0). Use the not equal to operator (`!=`). This will mean that every block the player passes over, except air will be changed.



For further exercises check out Python Programming with Minecraft Pi, the book available as a free pdf from www.arghbox.wordpress.com