

Python Pi Project #1: Teleporter

Level: Starter

Sometimes in Minecraft you can get lost. It would be nice if you had a way to get back where you started, quickly. Well here it is – a Python teleporter program.

When you start this program, it activates a teleporter that knows how to bring you back to the spot you're standing on. Now you can roam where you like – if you get lost you can teleport back to the starting point. How? You make a block of glass, and stand on it.

Coding skills needed

- variables:
`start =`
 - do things in a loop:
`while True :`
...
 - test a condition:
`if block_below == block.GLASS.id :`
...
 - using a module and objects:
`import mcpi.minecraft as minecraft`
`mc = minecraft.Minecraft.create()`
`start = mc.player.getTilePos()`
- Use object *properties*...
...and *methods*
- Create an object

Look out for the “dematerialize” special effect!

How does it work? First of all we import the modules we need to connect to minecraft. This is the same for pretty much any program that uses minecraft. We'll also get the 'time' module.

```
import mcpi.minecraft as minecraft
import mcpi.block as block
import time
```

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

Next we get the 'player' object from Minecraft and ask it for its position. This is going to be the position that the teleporter will jump to.

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

We want to let the player know the program has started so we use

Minecraft's 'postToChat' to display a message. We build a message showing the co-ordinates using Python's built in function 'str()', which turns pretty much anything into a 'string' of characters that can be joined into a message.

Then we start an endless loop. Every 0.3 seconds we work out co-ordinates for the block below us and get the 'id' for that block from Minecraft.

```
block_below = mc.getBlock(p.x, p.y - 1, p.z)
```

Then we check if the block id is the same as the id for 'GLASS' – in other words is the block below us made of glass?



If it is, we'll use the 'setBlock' function to change the player position back to where we started:

```
mc.player.setPos(start.x, start.y, start.z)
```

But before we do that, we put in a

'special effect'! We use 'setBlock' to fill the air around us with some glowing obsidian. All right, it maybe doesn't look too much like a teleport effect, but it's something!

The complete program

```
#!/usr/bin/env python

# A teleport program. Starting the program activates the teleporter
# with a destination where you are standing. To build a teleport pad
# make a block of glass and walk on top of it. This will teleport you
# back to where you started!

import mcpi.minecraft as minecraft
import mcpi.block as block
import time

mc = minecraft.Minecraft.create()

start = mc.player.getTilePos()
co_ords = str(start.x) + ", " + str(start.y) + ", " + str(start.z)
mc.postToChat("Teleport activated at " + co_ords)

while True:
    time.sleep(0.3)
    p = mc.player.getTilePos()
    block_below = mc.getBlock(p.x, p.y - 1, p.z)

    if block_below == block.GLASS.id :
        # teleport!
        print("Teleporting")
        mc.setBlock(p.x, p.y, p.z, block.GLOWING_OBSIDIAN)
        mc.setBlock(p.x, p.y+1, p.z, block.GLOWING_OBSIDIAN)
        time.sleep(1) # give us time to see it!
        mc.player.setPos(start.x, start.y, start.z) # jump
        # now clear the teleport again
        mc.setBlock(p.x, p.y, p.z, block.AIR)
        mc.setBlock(p.x, p.y+1, p.z, block.AIR)
```

Dojo Challenge #1:

See if you can make a better transporter special effect. What about a better launchpad than a block of glass?

Dojo Challenge #2:

Can you work out a way of putting the destination for the transporter at some particular location (not where you start)?