Python: Minecraft

Updating Raspberry Pi & Installing Minecraft

Update your RaspberryPi and install Minecraft.

```
$sudo apt-get update
$sudo apt-get install Minecraft-pi
```

Create Save & Exit *in Terminal*

File Extensions:	Creating File:	Save File in Terminal:	Exit & Rename:
.docx Microsoft Word .pdf PDF .py Python .js Java Script	Creating File (and ***NOT*** opening): touch filename.py Creating File (and opening in Terminal): nano filename.py	ctrl O	ctrl X *Follow prompts to rename if you want*

Terminal Directions:

ls (enter)	If the only thing listed is pi, then cd pi
	If there are several directories and Desktop is one of them listed, move to next step
cd Desktop (enter)	
cd Python_Minecraft (enter)	Some of you did not follow directions and did not name your Python_Minecraft folder as I did. You may have to double check this.
nano welcomeHome.py	This creates the new python program you will write

** Follow the same directions for the other files you will create

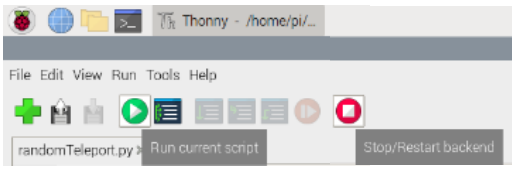
Running a Program from Terminal:

- Make sure you are in the Directory (folder) that your program is in. Use the following commands to get there:

Command	What it Does
ls	**L**ist**S** the files or directories in the current directory
cd _____	**C**hanges **D**irectory to the directory named
cd ..	Goes up one Directory

- Make sure you have Minecraft running and an open world.
- In the terminal, type:
  - **python3 filename.py**

Running a Program from Thonny:



- Make sure have Minecraft running and an open world.
- Open the program from the File System you want to run.
  - Press the Green Play Button across the top.
  - If you want to terminate the program, press the red Stop button across the top.

Minecraft:

Playing:

W- Move Forward

A- Move Left **S-** Move Right

D- Move Backward

Left Click- Break blocks
Right Click- Place blocks
E- Inventory
Space- Jump (Double tap to fly or stop flying)
Esc- Menu
Tab- Release Mouse Cursor

Import:

You can't call someone if they don't have a phone, in fact, you both need the tool of a phone to be able to communicate.

When you write a program in Python that will run in Minecraft, you have to "import" Minecraft.

The following command and the next command will be included in every Minecraft program. To import the Minecraft module to be used in your program you must type:

```
from mcpi.minecraft import Minecraft
```

Connect:

Even if two people have the tool of a phone to talk, one person has to dial and the other person has to answer. Just like one person has to call to talk, your program in Python has to "call" and make a connection to Minecraft.

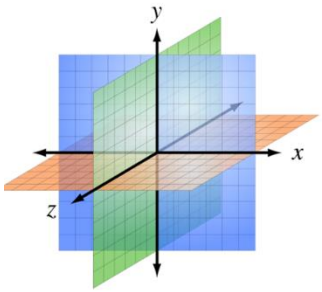
This line makes a connection to an ***open*** game. ***You must have your Minecraft game open and running for the program to work.***

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

Location:

Your player's coordinates can be found on the top left corner of your Minecraft screen.

You have graphed in math class with an x, y coordinate grid, however Minecraft is played in a 3D environment so everything in Minecraft uses x, y, and z to describe location.



How does x, y and z move the player in Minecraft?		(min, max)
x	East / West (left/right)	(-125,125)
y	Up/ Down	(-64,64)
z	North / South (forward/backward)	(-125,125)

Variables:

In Algebra, we are able to use variables to substitute values into an equation (see right). A variable is a way to represent and store information. In Python, we can set variables equal to numbers, strings (groups of letters and numbers) and functions (commands created from a series of other commands).

Simplify $2x - 3y$ if $x = 5$ & $y = -2$

$$2x - 3y$$
$$2(5) - 3(-2)$$
$$10 + 6$$
$$16$$

Look back on this paper and find an example of typed code that represents a variable. What makes it represent a variable like the example above where $x = 5$ & $y = -2$?

Code: mc = Minecraft.create() Key to making a variable: (equal sign) =

A common variable that we use is finding and using the player's position (pos for short).

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

Can now call your location using the x, y and z coordinates by using:

- `pos.x`
- `pos.y`
- `pos.z`

Another way to use the x, y and z coordinates is by setting the variables with Python's unpacking technique:

```
x, y, z = mc.player.getPos()
```

Can now call your location using the x, y and z coordinates by using:

- x (calling the x value of the player's position)
- y (calling the y value of the player's position)
- z (calling the z value of the player's position)

API: Application Programming Interface

An API is a set of subroutine definitions, communication protocols and tools for building a program. You'll want to Google an API for Minecraft to find the different codes for the blocks!

Pseudocode:

Pseudocode is an informal way of describing programming language syntax without using programming language. Is later converted into programming language. Like the diagram we saw programming with Scratch, it helps a programmer think through what they are trying to program.

Pros	Cons
Allows you to logically think through and explain what you want your code to do before writing it.	Cannot bet tested or debugged, hence you might have missed a step.

Let's talk through some Pseudocode for a few programs and then use the API to try to write the code.

helloWorld.py

Pseudocode	Python Code
Description of what the program does.	# Posts the message "Hello World" on chat in Minecraft
Import Minecraft	`from mcpi.minecraft import Minecraft`
Connect to Minecraft	`mc = Minecraft.create()`
Post Message to Chat Screen	`mc.postToChat("Hello World")`

whereAmI.py

Pseudocode	Python Code
Description of what the program does.	# Posts the player's position every 5 seconds in Minecraft's chat
Import Minecraft	`from mcpi.minecraft import Minecraft`
Connect to Minecraft	`mc = Minecraft.create()`
Find the x, y and z coordinate of the player's position.	`pos = mc.player.getPos()`
Post the Player's x, y, z coordinates to the Chat Screen	`mc.postToChat("Player Position; x: " + str(pos.x) + ", y: " + str(pos.y) + ", z: " + str(pos.z))`

If you check the top left of the screen, these values should match the values written in white.

stackingBlocks.py

Pseudocode	Python Code
Description of what the program does.	# The program identifies the player's position and stacks melons vertically at the player's position
Import Minecraft	`from mcpi.minecraft import Minecraft`
Make a connection to Minecraft	`mc = Minecraft.create()`
Get the player's position	`pos = mc.player.getPos()`

Set the x,y,z coordinates of the player's position to a variable so that it's easier to call.	`x = pos.x`
	`y = pos.y`
	`z = pos.z`

Set the block type we will use to block 103 which is a melon	`blockType = 103`
Set the block at the player's position to a melon	`mc.setBlock(x+1,y,z, blockType)`
GOING UP by one,	`y = y+1`
Set a block on top of the last melon placed.	`mc.setBlock(x+1,y,z, blockType)`
GOING UP by one,	`y = y+1`
Set a block on top of the last melon placed.	`mc.setBlock(x+1,y,z, blockType)`
GOING UP by one,	`y = y+1`
Set a block on top of the last melon placed.	`mc.setBlock(x+1,y,z, blockType)`
GOING UP by one,	`y = y+1`
Set a block on top of the last melon placed.	`mc.setBlock(x+1,y,z, blockType)`

Challenge: Try creating a loop to condense this code to build a stack of melons 4 high.

blockBeneath.py

Pseudocode	Python Code
Description of what the program does.	`# The program places the block beneath the player to glass`
Import Minecraft	`from mcpi.minecraft import Minecraft`
Make a connection to Minecraft	`mc = Minecraft.create()`
Get the player's position	`pos = mc.player.getPos()`

Set the x,y,z coordinates of the player's position to a variable to make it easier to call.	`x = pos.x`
	`y = pos.y`
	`z = pos.z`

Set the block type we will use to block 20	`blockType = 20`
Decrease the y-value by 1, so that it identifies the block beneath the player	`y -= 1`
Now, that we've decreased the y value by one, this will set the block below the player to glass.	`mc.setBlock(x,y,z, blockType)`

Challenge: Can you get program to constantly run while playing? Can you postToChat what the block is and what the program is changing it to? Can you check what block is beneath you and only change it if it's a certain type of block?

speedBuilding.py

Psuedocode	Python Code
Description of what the program does.	`# The player will build a hollow building with the program`
Import Minecraft	`from mcpi.minecraft import Minecraft`
Make a connection to Minecraft	`mc = Minecraft.create()`
Get the player's position	`pos = mc.player.getPos()`

Set the x,y,z coordinates of the player's position to a variable to easily call later.	`x = pos.x`
	`y = pos.y`
	`z = pos.z`

Set the block type we will use to block 4 which is cobblestone	`blockType = 4`

Determine the width(x), length(z) and height(y) of the building	`width = 10`
	`length = 12`
	`height = 5`

This will set all the blocks in width by length by height to cobblestone	`mc.setBlocks(x,y,z, x + width, y + height, z + length, blockType)`
But I don't want the inside to be solid, so I have to set everything one inside from the outer wall to be air.	`mc.setBlocks(x+1, y+1, z+1, x + width - 1, y + height - 1, z + length - 1, 0)`

** Students will probably type in mc.setBlock (and forget the s on Blocks) for the last program.

Challenge: How can you fix this code so the building doesn't build right on top of you? Can you add to the code so that there is an opening for a door and you don't have to break the blocks to make the door?

Last night you were asked to create a house in a world in Minecraft. Go into the world and identify the x, y, z coordinates of the front door of your house.

$x =$ _____

$y =$ _____

$z =$ _____

Using your unique x, y, z coordinates, create a new file and write the following program:

welcomHome.py

Psuedocode	Python Code
Description of what the program does.	`# The program will display "welcome home" in the player chat when the person is at the front door of their home they created.`
Import Minecraft	`from mcpi.minecraft import Minecraft`
Make a connection to Minecraft	`mc = Minecraft.create()`
Import Time	`import time`

Every second it will check if the player is standing at the specific x, y, z coordinates and if TRUE, it will post "Welcome Home" to the player's chat.	`while True:`
	`  time.sleep(1)`
	`  pos = mc.player.getTilePos()`
	`  if pos.x == ____ and pos.y == ____ and pos.z == __:`
	`  mc.postToChat("Welcome Home")`

After writing the program, run the program (python welcomeHome.py) with your player, you need to move away from your house and move back to the front door and tell me what happens below.

randomTeleport.py

Psuedocode	Python Code
	`# The program will teleport the player to random positions in their current world 5 different times and displaying the x, y, z coordinates where they land.`
Import Minecraft	`from mcpi.minecraft import Minecraft`
Import random	`import random`
Import Time	`from time`
Make a connection to Minecraft	`mc = Minecraft.create()`
Create a variable to keep track of how many times you go through the loop	`count = 1`

Checks the *variable count* and if it is less than 6 (so 1,2,3,4,5) then it will proceed with the given code, creating a random location based on parameters given. Notice that the y value isn't -64 to 64. We can't just go anywhere and put the player underground- but is likely to still happen since this isn't a flat world.	`while count < 6:`
	`  x = random.randint(-127, 127)`
	`  y = random.randint(0, 64)`
	`  z = random.randint(-127, 127)`

Sets the players position to the random x, y, z numbers generated in lines above	`  mc.player.setTilePos(x,y,z)`
Post to chat how many times the player has teleported and the x, y, z coordinates to where they teleported to.	`  mc.postToChat("You've teleported " + str(count) + " times and landed at x= " + str(x) + ", y= " + str(y) + ", z= " + str(z))`
Increase the variable count by 1 each time the code goes through the loop. Once it is 5, (so count runs the program when count is 4, then add one, it will exit the loop and the program is done)	`  count += 1`
Makes the program wait 5 seconds before checking the loop again, so you can see your player being moved to the different places rather than it happening as a flash.	`  time.sleep(5)`

Challenge: Create a condition for the first teleportation so that your PostToChat comment reads "You've teleported 1 **time** and landed at …" rather than "You've teleported 1 **times** and landed at…." Create a condition that checks to see if you are underground with the random numbers chosen.