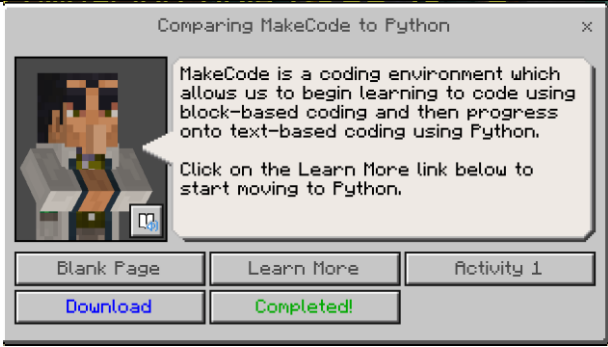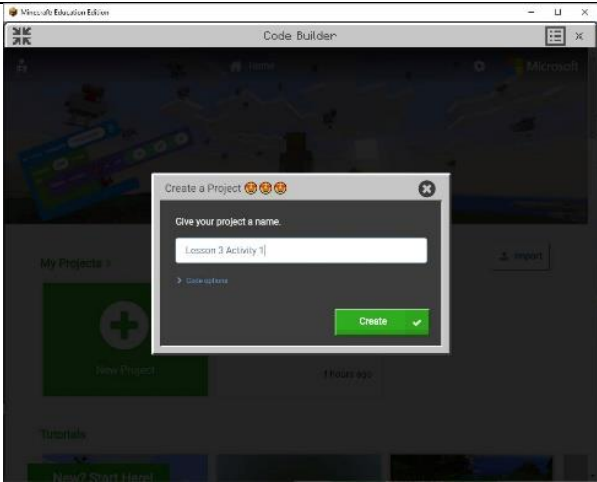## Coding in Minecraft – Advanced Coding Using Python
Lesson 3 – Python – Comparing MakeCode to Python
Activity

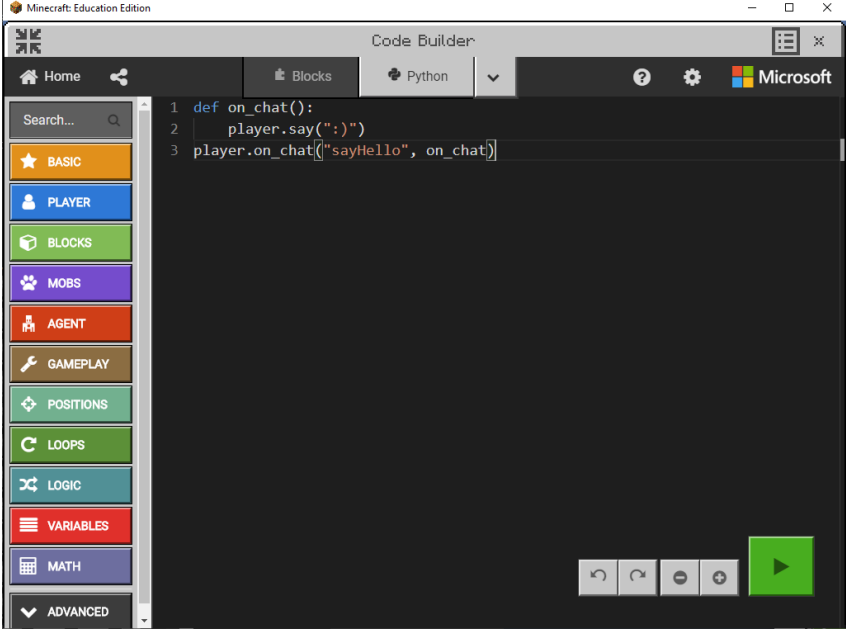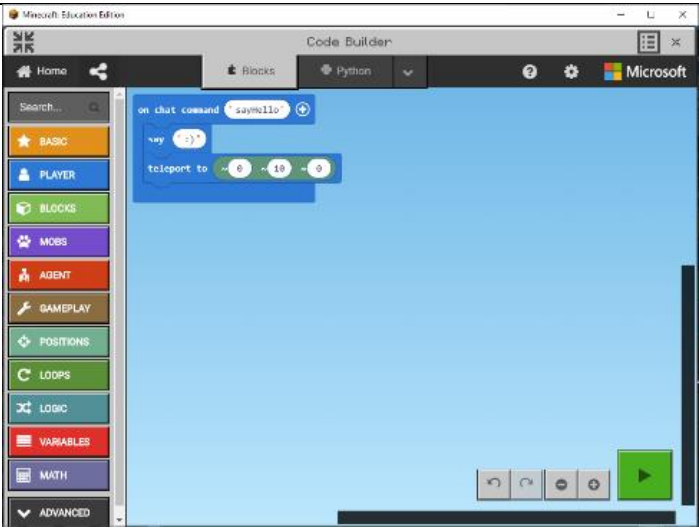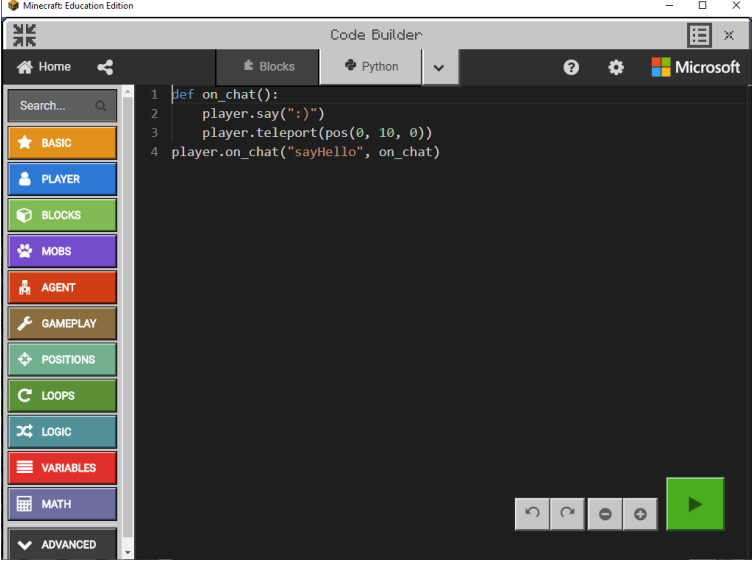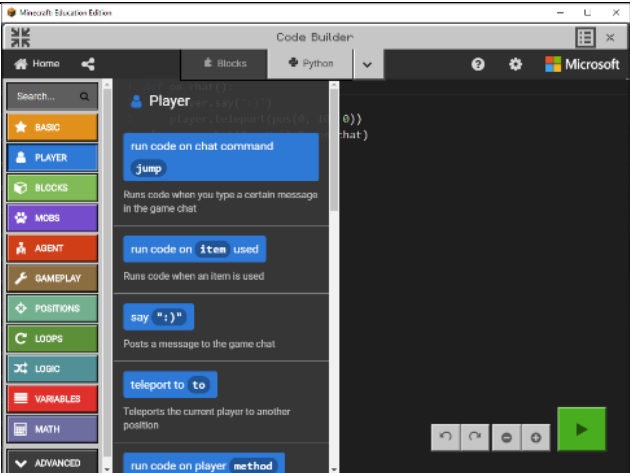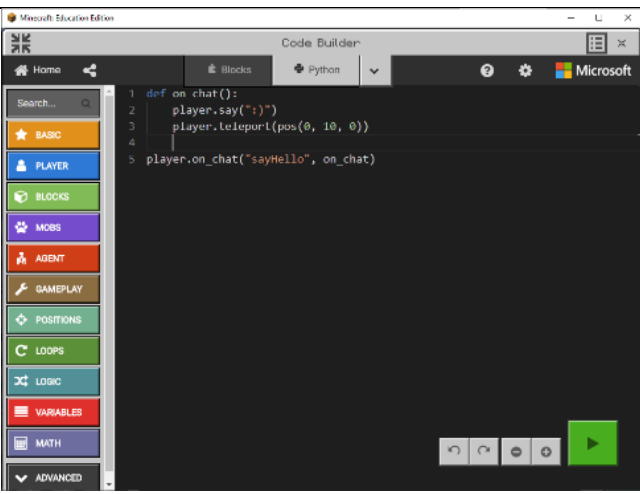**Instructions**

This worksheet is copy of the steps detailed in the Activity in the game to make it easier for you to follow along and try yourself.

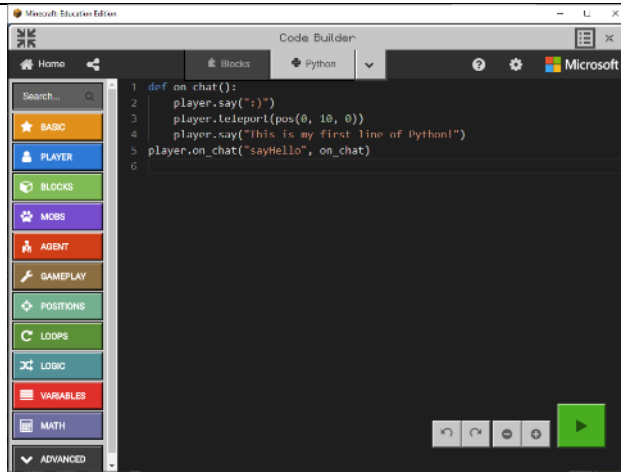| Step	Instructions
1	In this activity we will explore the Python interface in Code Builder and compare various MakeCode blocks to Python code. **Read through the whole tutorial and then try to perform the actions**
2	In the Minecraft world click on the Blank Page button in the NPC labelled Comparing MakeCode to Python.
3	Click on the New Project icon to create a new MakeCode project When prompted set the project name to Lesson 3 Activity 1.

4		The default block-based coding editor with the default on chat command "run" block and on start block will appear. Change the event handler to respond to the command sayHello by changing the text run to sayHello
5		Click on the Convert code to Python button (you may need to click on the down arrow to change from JavaScript to Python) to display the code in Python.
6	Notice there are 3 lines of code. The first line of code starts with def on_chat - this is creating a function called on_chat. The 2nd line of code is indented - this means that the line of code is part of the function on_chat. The line of code pass does nothing - pass is like a placeholder which you replace with code later. The 3rd line of code is setting the function on_chat as an event handler for the event on_chat of the player when the chat words are sayHello. The first part within the brackets specifies the chat command to respond to and the second command specifies which function to run when that chat command is entered. when we build up what we want to happen when sayHello is issued.		

7		Switch back to the block-based view by clicking on the Convert code to Blocks button. Add a player say ":)" block from the PLAYER drawer into the event handler.
8	Change back to the Python view. Notice that a single line of code has been added at line 3 - this is the equivalent of the player say block.	

9		Switch back to the block-based view by clicking on the Convert code to Blocks button. Add a player teleport to ~0 ~0 ~0 block from the PLAYER drawer into the event handler and change the Y coordinate to 10
10	Change back to the Python view. Notice that a single line of code has been added at line 3 - this is the equivalent of the ``||player: teleport to ~0 ~0 ~0 ||`` block. Again the teleport block came from the player drawer so the code starts with the word player and we want the player to be teleported therefore the teleport function is called. This function takes a parameter - where to be teleported to. This parameter is a coordinate.	

11		Whilst in Python view you can choose to either enter text into the editor window or drag blocks of Python from the drawers like you did with MakeCode blocks
12		Add a new empty line to your code after line

player.teleport(pos(0, 10, 0))

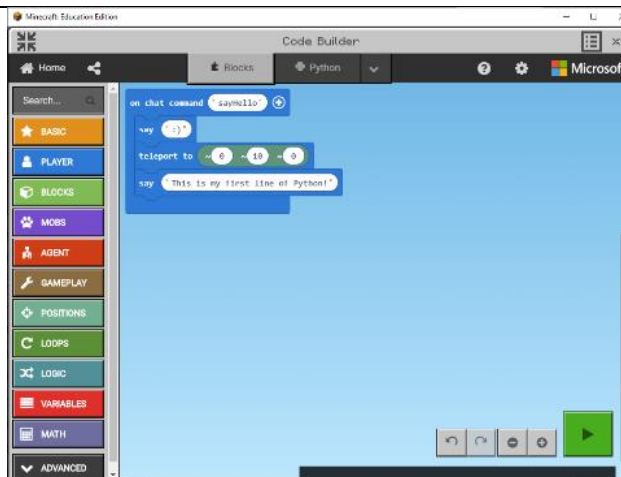by placing your cursor at the end of that line and pressing the Enter key. |

| 13 | | Press the tab key on your keyboard to indent the line of code (to add it to the function) and enter the Python code
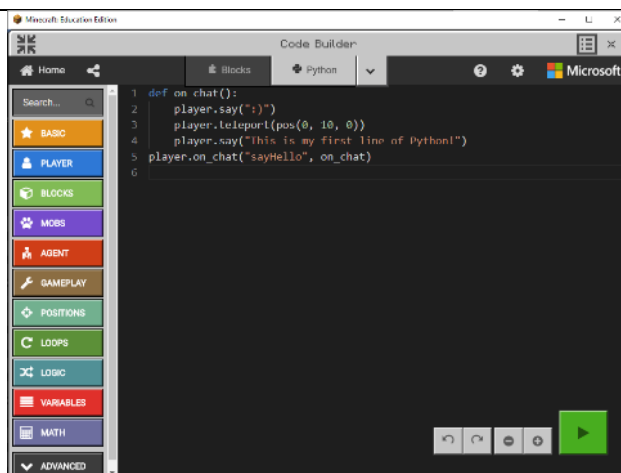
player.say("This is my first line of Python!")

on the new line. |
| 14 | | Switch back to the MakeCode block view and notice there is now a player say "This is my first line of Python!" block. |
| 15 | | Switch back to Python. |

| 16 | Create another empty line after the code

player.say("This is my first line of Python!")

you just entered.

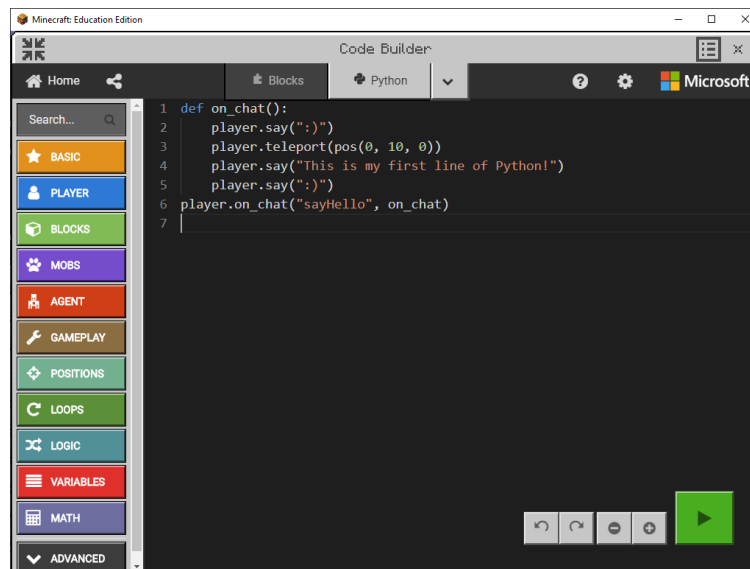Open the PLAYER drawer and notice there are a number of blocks just like in MakeCode blocks view.

Find the block player say ":)" and drag it out and place it on the recently created empy line in your code.

You should notice the code

player.say(":)")

has been added. |

| | It can be tricky to position the Python block easily so its best to just type Python text into the editor.

 |
| 17 | Return to the game and move onto the next Non Player Character |