



Python Coding in Minecraft: Education Edition with Microsoft Azure Notebooks

Instructional Support Guide

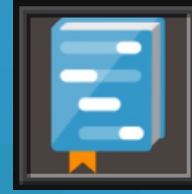




- Educator Crafting Table

MINECRAFT

EDUCATION EDITION

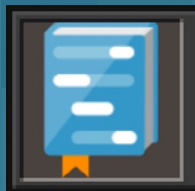


Contents

- 00 Setting Up the Coding Window
- 01 Introducing Python Coding in the Notebook Editor
- 02 Get Coding with Tutorials
- 03 Python Variables in Notebooks
- 04 Syntax of Coordinates with World Set Command in Notebooks
- 05 Python Event Handlers in Notebooks

MINECRAFT

EDUCATION EDITION

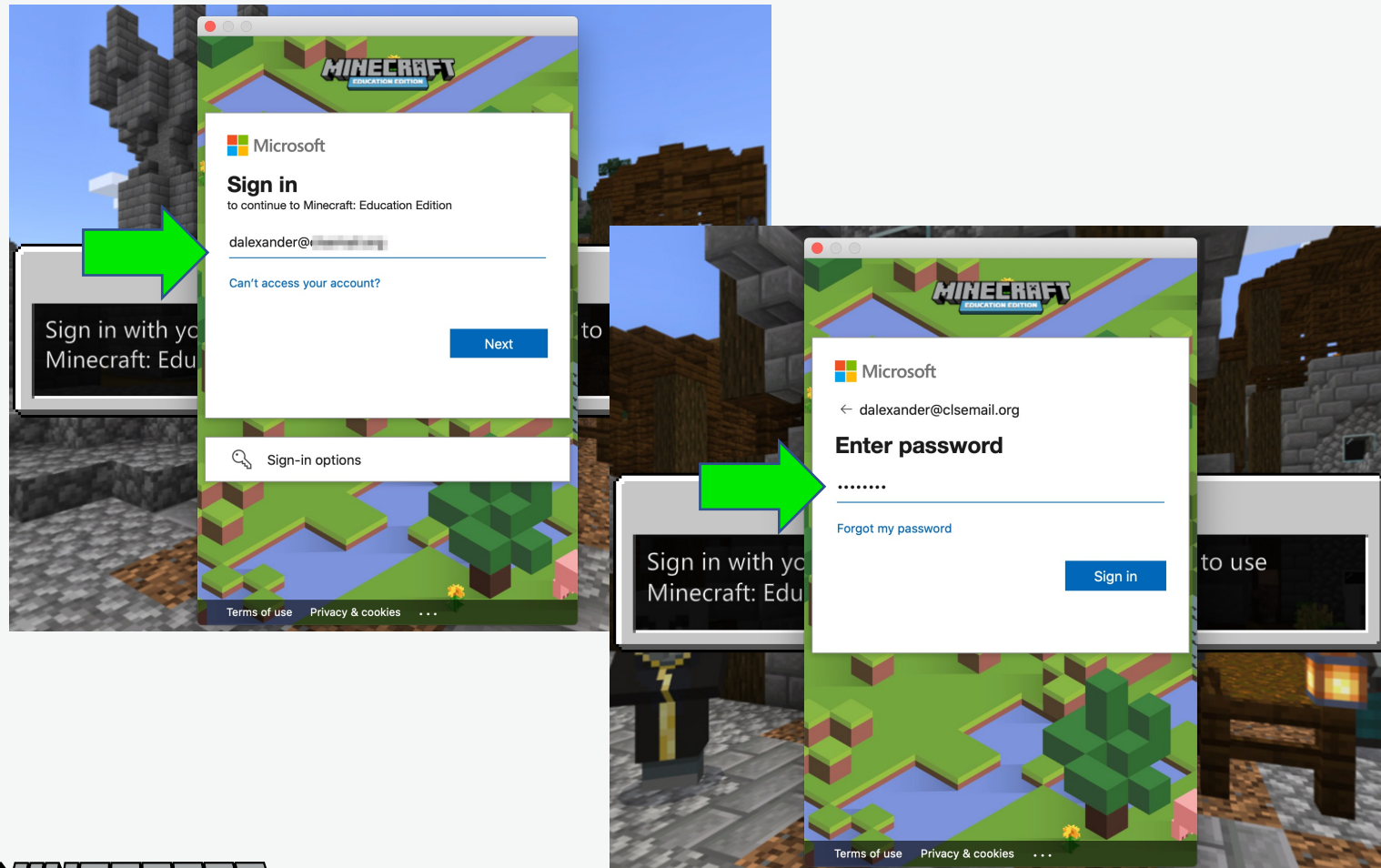


00

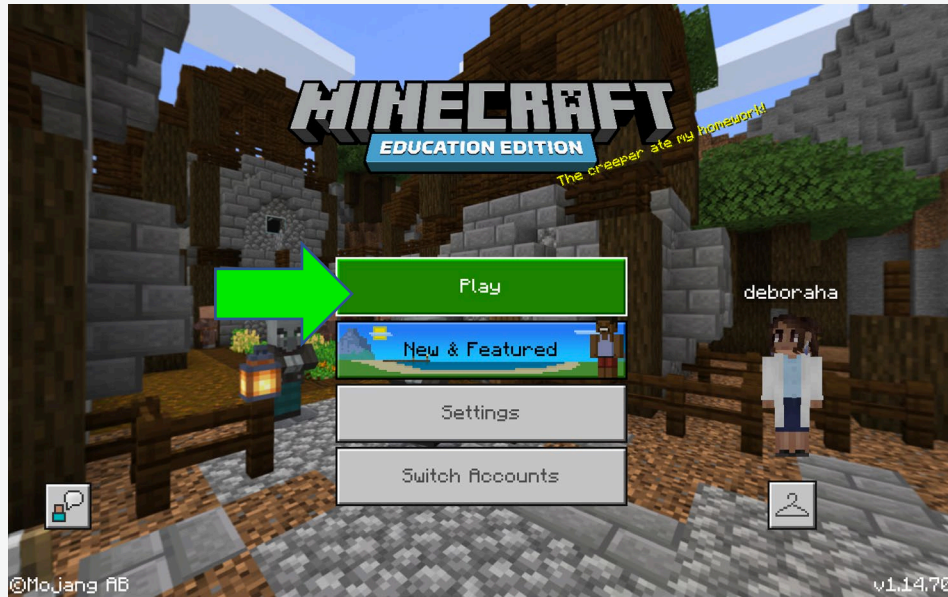
Let's Get Set UP



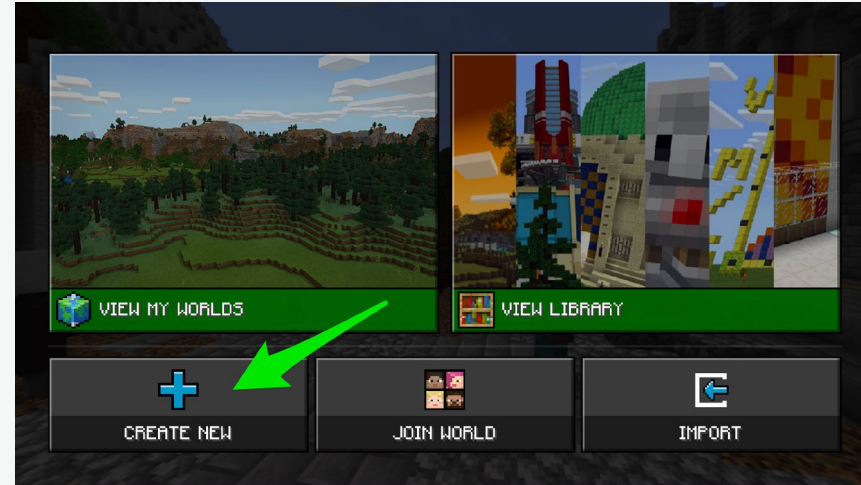
Sign in with your education account and password



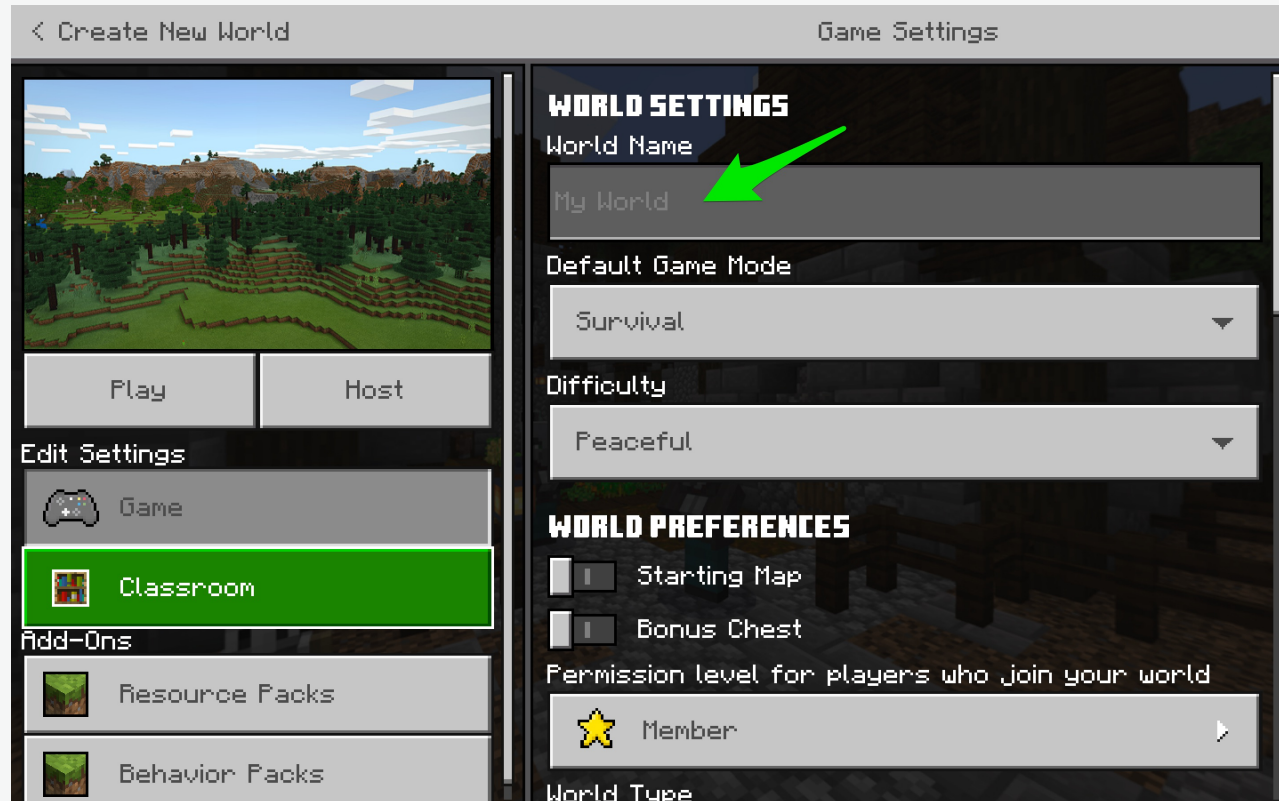
Create New World



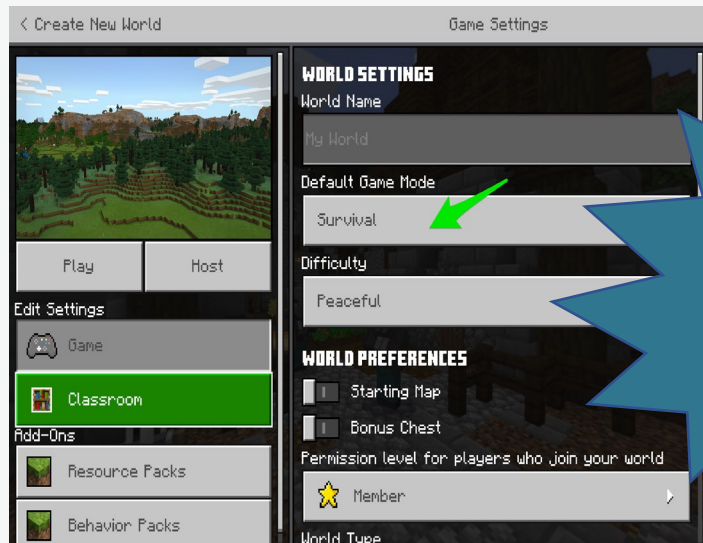
There are lots of options, but for now, create new.



Give your world a good name!

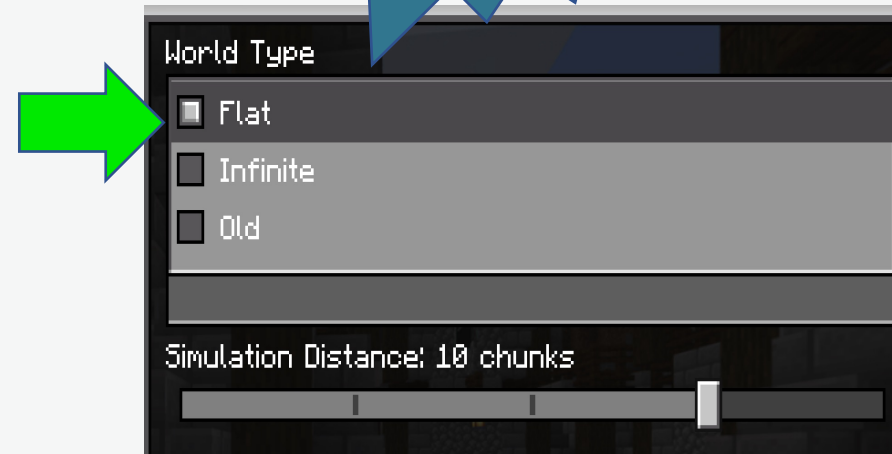


Use the settings best for your lesson



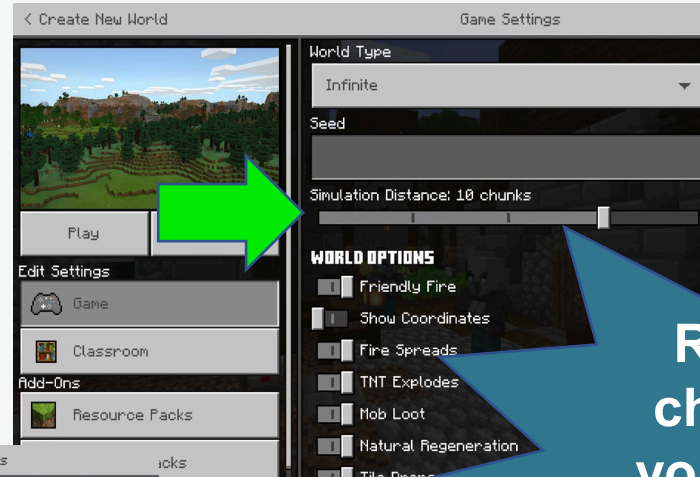
Creative,
Peaceful

Flat
worlds are
easy to
code!

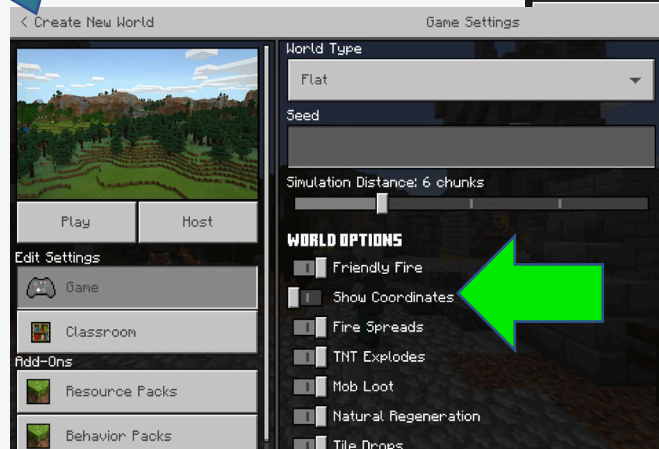


Use the settings best for your lesson

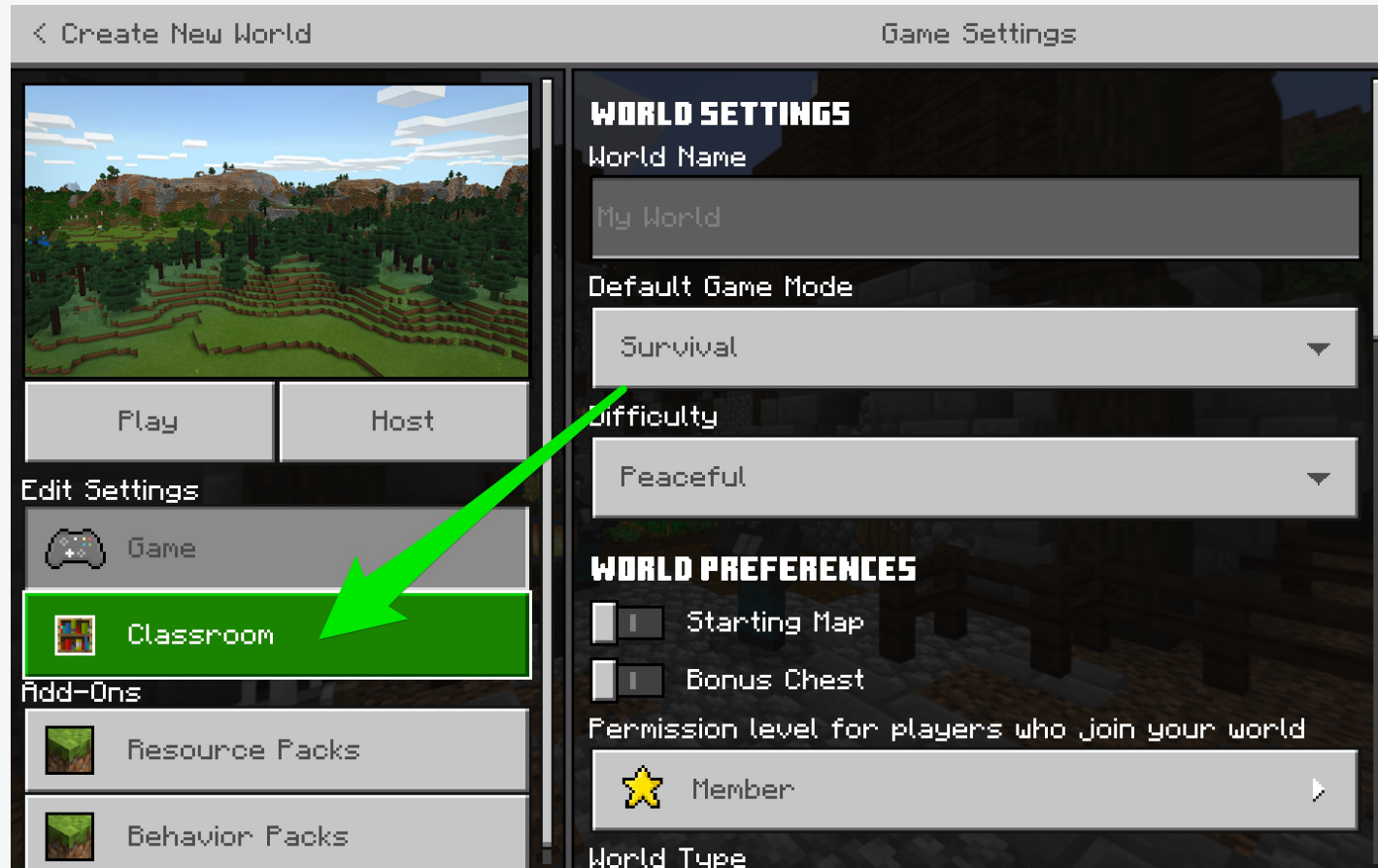
Show coordinates
- SO
important!



Reduce chunks if
your game
lags



New Classroom Settings



New Classroom Settings



The screenshot shows the 'Classroom Settings' menu in Minecraft Education Edition. The left sidebar contains options for 'Play', 'Host', 'Edit Settings', 'Game', 'Classroom', 'Add-Ons', 'Resource Packs', and 'Behavior Packs'. The main settings list includes:

- Allow Commands
- Code Builder
- Always Day
- Perfect Weather
- Keep Inventory
- Allow Mobs
- Allow Destructive Items
- Player Damage
- Player VS Player Damage
- Immutable World
- Show Border Effect

Two green arrows point to the 'Always Day' and 'Perfect Weather' settings. A blue callout box on the right contains the text: **Perfect weather and always day are less distracting!**

RESOURCE LINK
Create a new button on the Pause menu that links to an assessment tool,



New Classroom Settings

< Create New World Classroom Settings



The screenshot shows the 'Classroom Settings' menu in Minecraft. On the left, there are navigation buttons for 'Play' and 'Host', and a section for 'Edit Settings' with options for 'Game' and 'Classroom'. Below that is an 'Add-Ons' section with 'Resource Packs' and 'Behavior Packs'. The main area on the right contains several toggle switches: 'Allow Destructive Items', 'Player Damage', 'Player VS Player Damage', 'Immutable World', and 'Show Border Effect'. Below these is a 'RESOURCE LINK' section with a description: 'Create a new button on the Pause menu that links to an assessment tool, website, or other resource for your world.' It includes two input fields: 'Resource Button URL' (with a placeholder 'Required to enable button') and 'Resource Button Name' (with a placeholder 'OPEN LINK (Default)'). A green arrow points to each of these input fields. At the bottom of the settings area is a link that says 'Get help with Classroom Settings'.

Play Host

Edit Settings

Game Classroom

Add-Ons

Resource Packs Behavior Packs

Allow Destructive Items

Player Damage

Player VS Player Damage

Immutable World

Show Border Effect

RESOURCE LINK

Create a new button on the Pause menu that links to an assessment tool, website, or other resource for your world.

Resource Button URL

Required to enable button

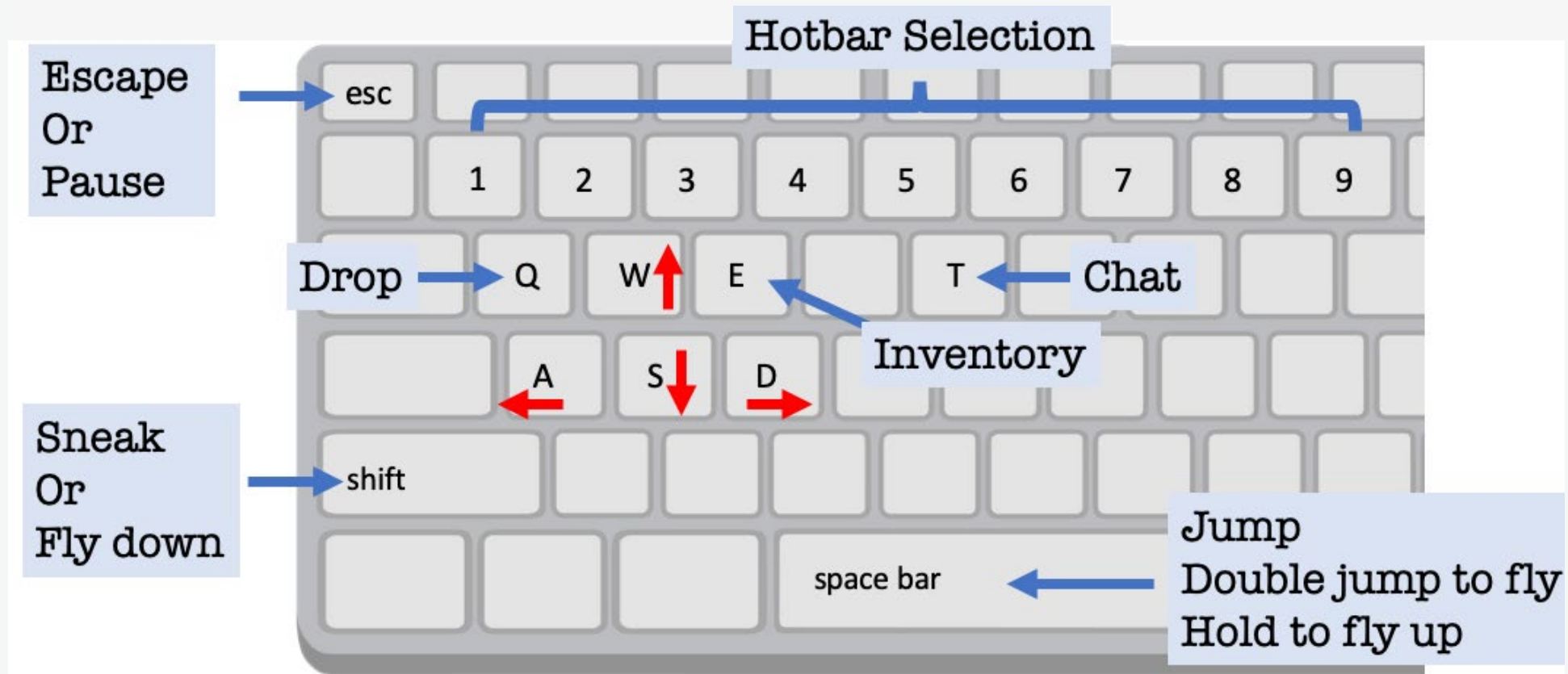
Resource Button Name

OPEN LINK (Default)

[Get help with Classroom Settings](#)

You can enter any link (maybe instructions!) and customize for your students.

The Keyboard in Minecraft: Education Edition




Try it out!

TUTORIAL 1. MOVEMENT

Language: English

Video Training Downloadable Materials

Beginner



Submitted by: [Minecraft Education](#)
Updated: July 1, 2020

[Share or assign](#)
[Print](#)

TUTORIAL 1. MOVEMENT

This Tutorial Will Teach You How To Navigate.

This tutorial will teach you how to navigate in Minecraft. Using W A S D keys and a mouse will help you move in Minecraft.
You will learn: - To walk - To jump - To turn around - To sprint

Media References

Movement Tutorial
To run, launch the world. From within Minecraft choose

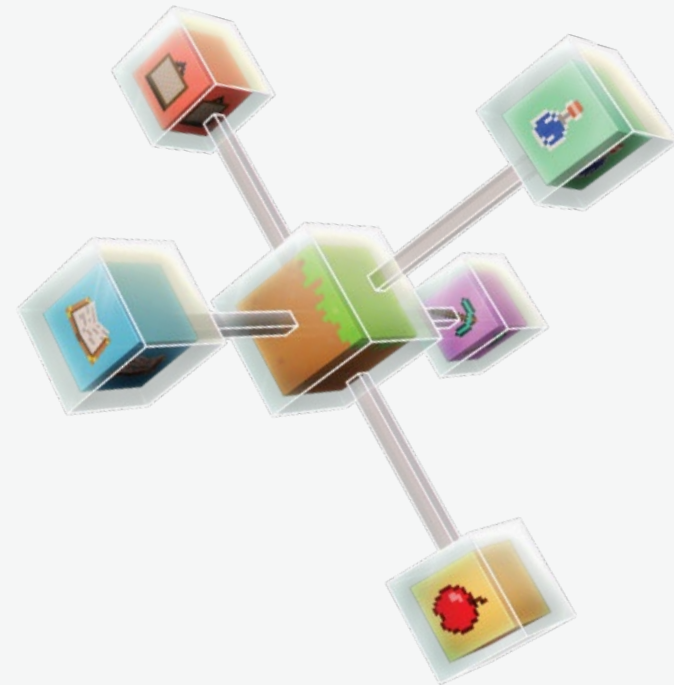
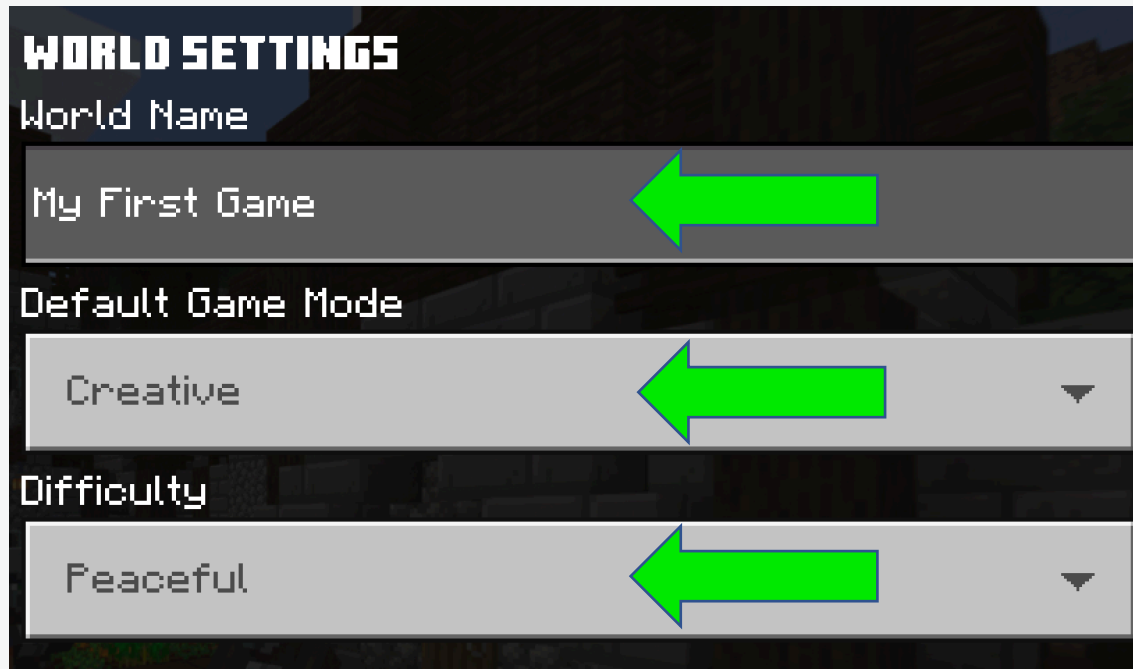
- Play/Create New/Templates.



[Tutorial 1. Movement | Minecraft Education Edition](#)

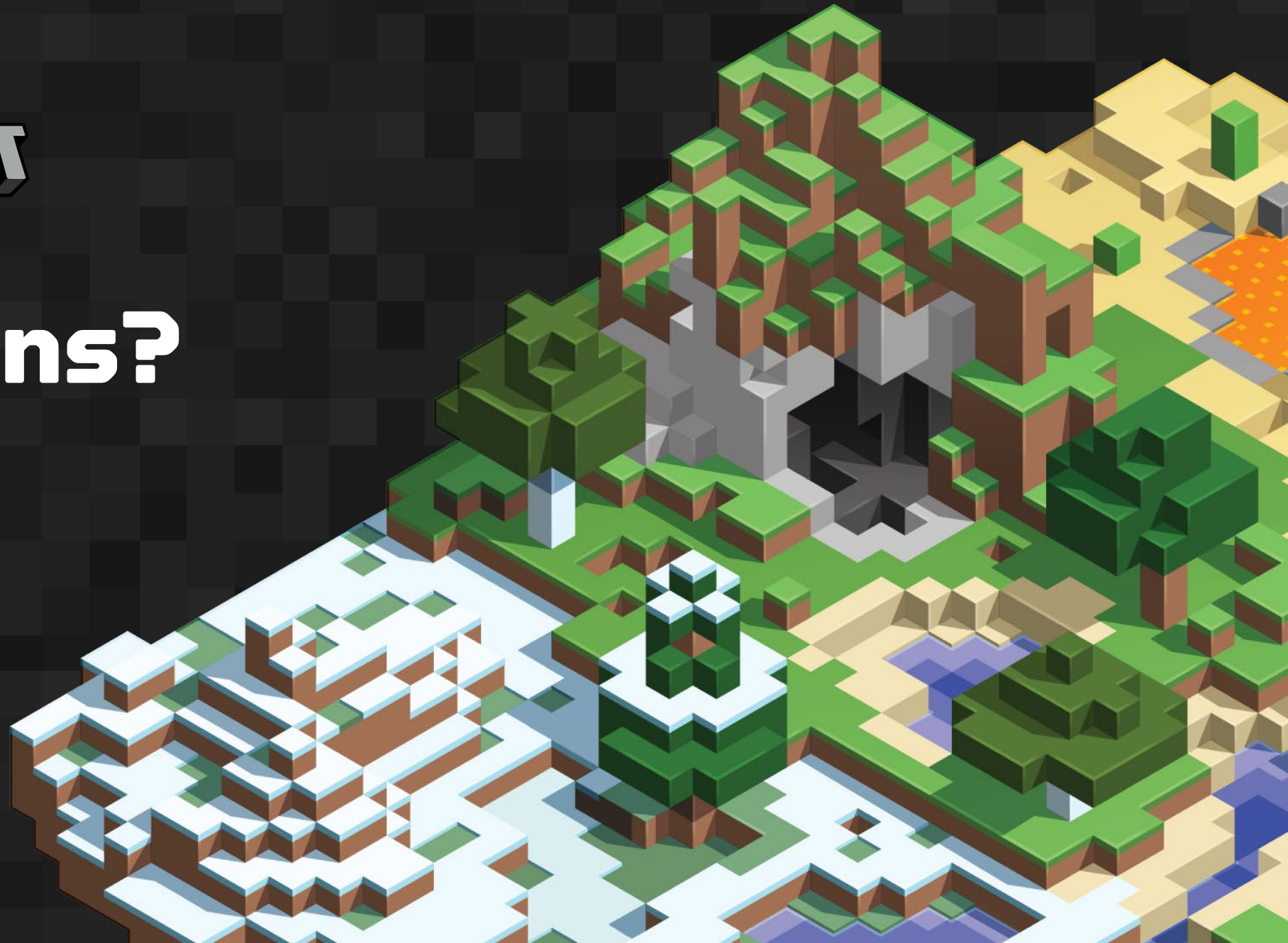


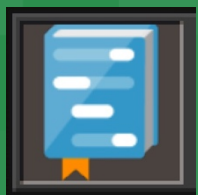
Suggested “go-to” settings for your first game



MINECRAFT
EDUCATION EDITION

Questions?





01

First Lesson to Introduce Python Coding in the Notebook Editor in Minecraft: Education Edition



First Steps into Python Coding with Minecraft in Notebooks



Yes, you CAN code in Python, and yes, your students will love to code in Minecraft: Education edition!

Here is a *first step* into actual Python code that your students can create and execute right in their own Minecraft world.

This is an introductory lesson.

- You do not have to do all of these commands, or you can start with these and do more.

Be flexible! Be brave! Get started and get coding!

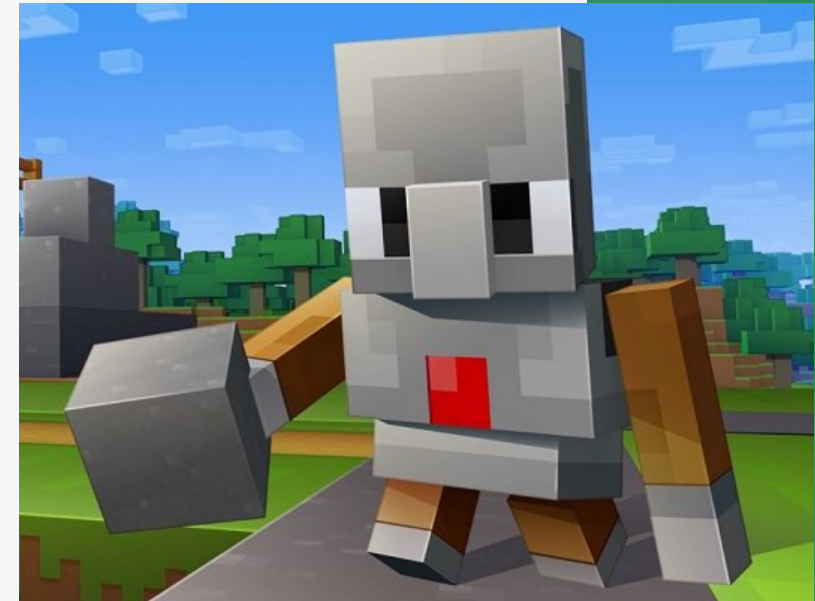
Building on success

The best way to code is from success to success.

- Start with one line of code that you know works. Run it. Be successful.
- Then edit it and see if you can make an edited line of code that works. Once you make sure that first line of code is working, add a second line. In other words, build on success.
- Fixing errors is part of coding. Fixing errors after adding three lines of code is ten times harder than fixing errors after adding one line of code.
- So keep changes small and you can get things fixed pretty quickly.
- Try not to worry about errors. Roll with the punches.

Computing is learning.

It just takes time to get over the bumps.



Growth Mindset!

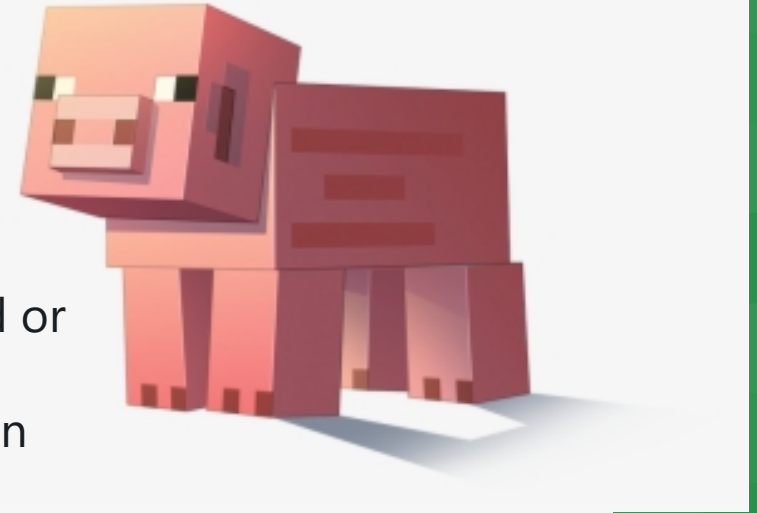
Computers are picky, picky picky. You cannot ignore a comma, period or lower case letter. For example, if you have one little space at the beginning of a line, or you use a "special" character, computers can get upset. They need all the details to be exact.

So look at every single thing you typed if you find you made an error.

Don't worry; every coder makes mistakes and learns from them!

Have fun!

You are in the driver's seat on an amazing adventure. You are creating things where usually you are just going along for the ride. Instead of looking at other people's art, you are the artist. Enjoy the process. Realize that you learn as much from the mistakes as you do from the successes.



Code Follows



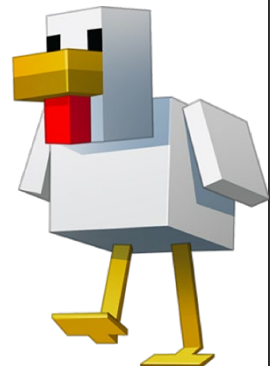
Teachers, students should begin their code in a flat world.

- Students should not begin by typing all of this code.
- Begin with a single command, and test.
- Edit the command and test.
- Make another version of the command, and guess what?
- Test.
- Add new commands after they have succeeded.

For example, after saying Hello World, they should change the command to their own message, and add an additional message of their own. Other edits could include summoning other animals, for example, a parrot. Try making small edits to the teleport command.



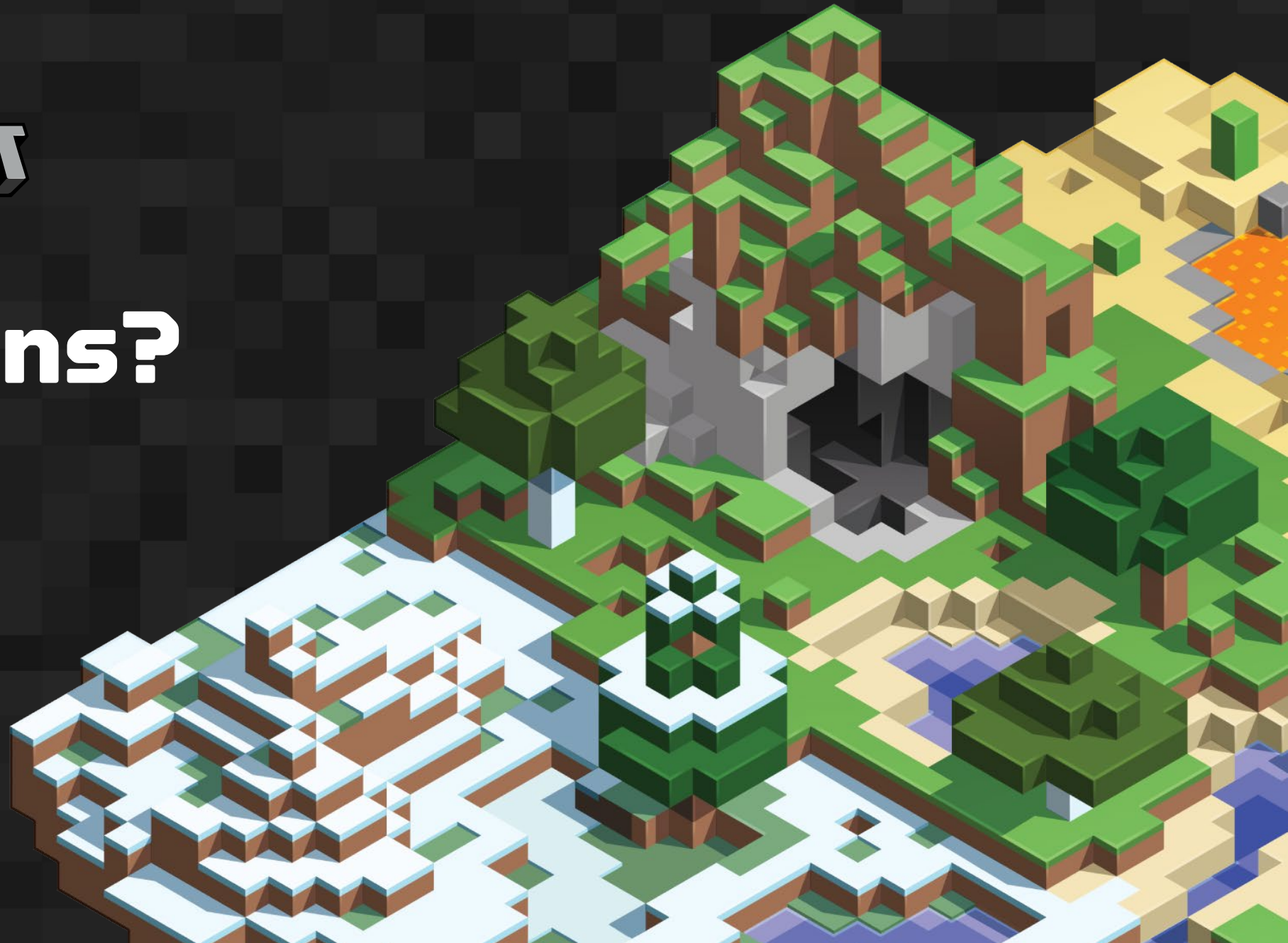
```
# First Lesson Commands: Comment, Say, Teleport, Summon
# This is a comment; comments begin with a hashmark
# Comments are very important and should be in every program
# Comments do not execute - they are notes for programmers.
#
# All first programming endeavors should begin with Hello World!
say("Hello World")
#
#
# Summon is like spawn. Names must be exactly correct.
summon("mooshroom")
#
# This is a teleport command. You can edit the 3 numbers.
# !!! This is meant for a FLAT WORLD !!!
player.teleport("10 5 10")
#
# What do these three commands do?
# Let's make sure the animal survives!
say("Looking up!")
player.teleport(from_me( 0,25,0))
summon("chicken")
#
# Ready for more? Here is more complex coordinate use!
world.fill("2 ~ 2", "5 3 5", "stone")
```

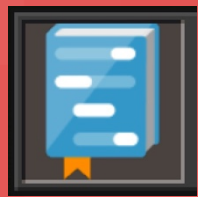


MINECRAFT

EDUCATION EDITION

Questions?





02

Get Coding with Tutorials

Billboards

Show_title(title, subtitle, action bar)

- One of the simplest tutorials in the Python notebook editor is **Billboards**.
- It uses the command *show_title*.

This simple command has very exciting results.

Recall previous experience using the command say:
say("Hello World!")

The say command required a command name: **say**, parens **()** and a string in quotation marks: **"Hello World!"**

In the show_title command, we enter our own information in the quotation marks, as well.



Billboards

Put your own words in this command.
`show_title("Mooshrooms are great!")`

Remember that computers are picky. You cannot skip the underscore.

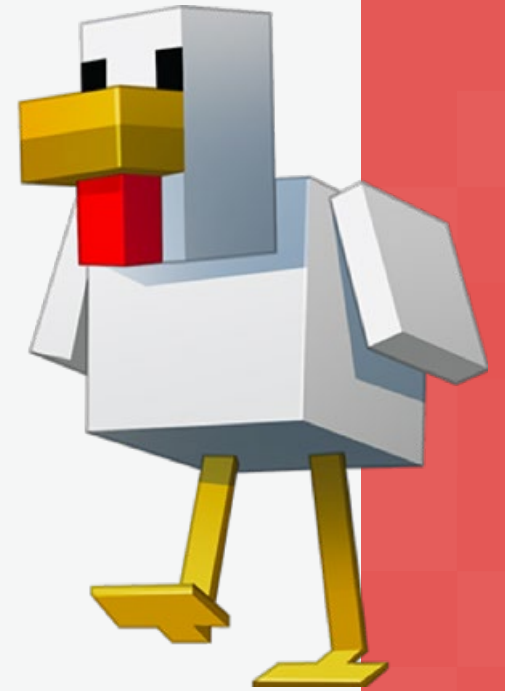
You cannot type:

show title with a space and no underscore. You cannot type **showtitle** with no space.

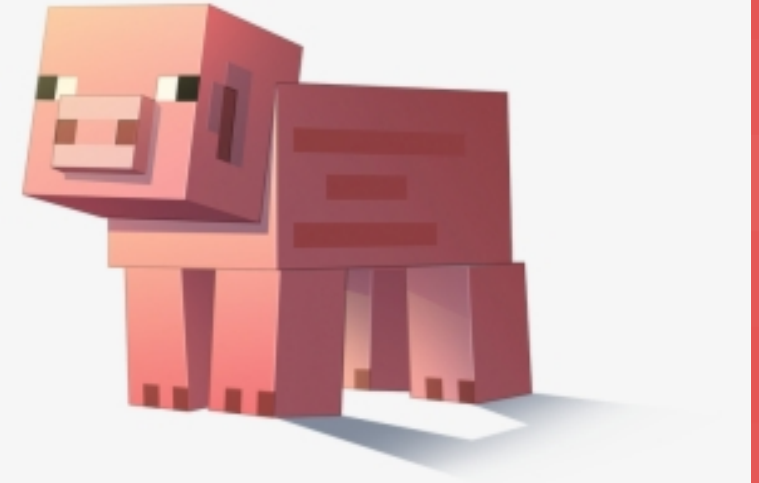
Students can edit the string to create different titles.

Then they can go on to adding a subtitle. The title and subtitle are two parameters, separated by a comma.

Interesting observations can be made about what happens if you “accidentally” forget the comma.



Billboards



What kind of error is produced if you miss the comma
`show_title("Coding", "is the best")`

A third parameter is allowed.

As a hint, have students keep the text short.

Going to three parameters requires careful error checking.

Check quotation marks and commas.

Don't forget that last paren!

Billboards

Be careful with syntax when adding the third parameters, for the action bar
`show_title("Coding!", "Lets me be in charge.", "Show me the commands!")`



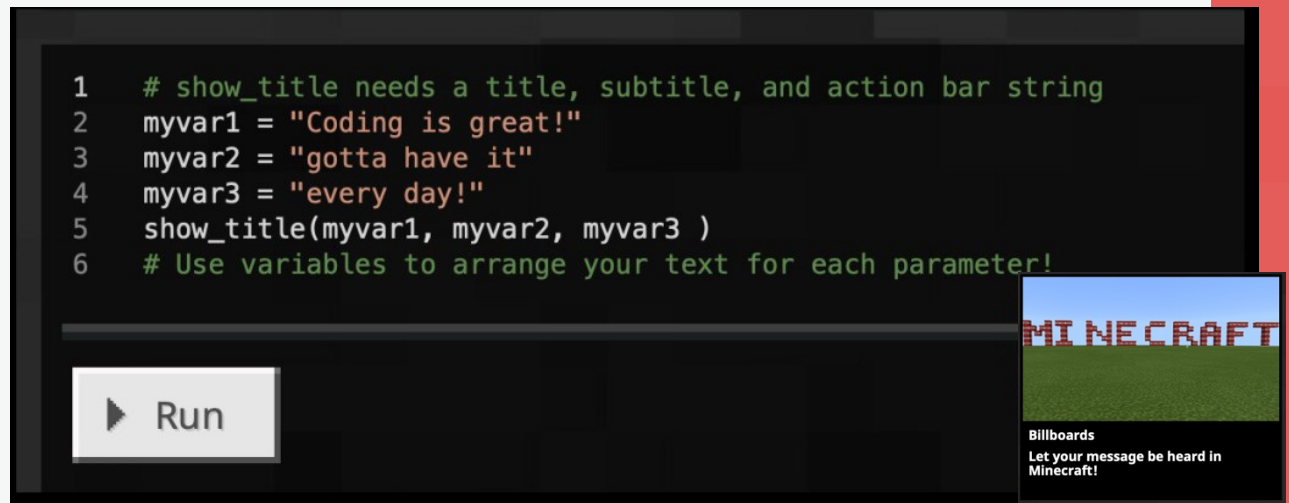
Extension Activity - Variables

- As an extension, you can use this command with variables.
- If your students are comfortable with variables, or if you have already tried the introduction to variables, you can add another level to your lesson.

The extensions offered in the tutorial in-game add even more parameters. Go ahead and use the Notebooks Tutorial in the game now.

As a hint: It is suggested that students do not experiment with large numbers for these settings. =] Go coding!

```
1 # show_title needs a title, subtitle, and action bar string
2 myvar1 = "Coding is great!"
3 myvar2 = "gotta have it"
4 myvar3 = "every day!"
5 show_title(myvar1, myvar2, myvar3 )
6 # Use variables to arrange your text for each parameter!
```

The screenshot shows a code editor window with a dark background and light-colored text. The code is as follows:

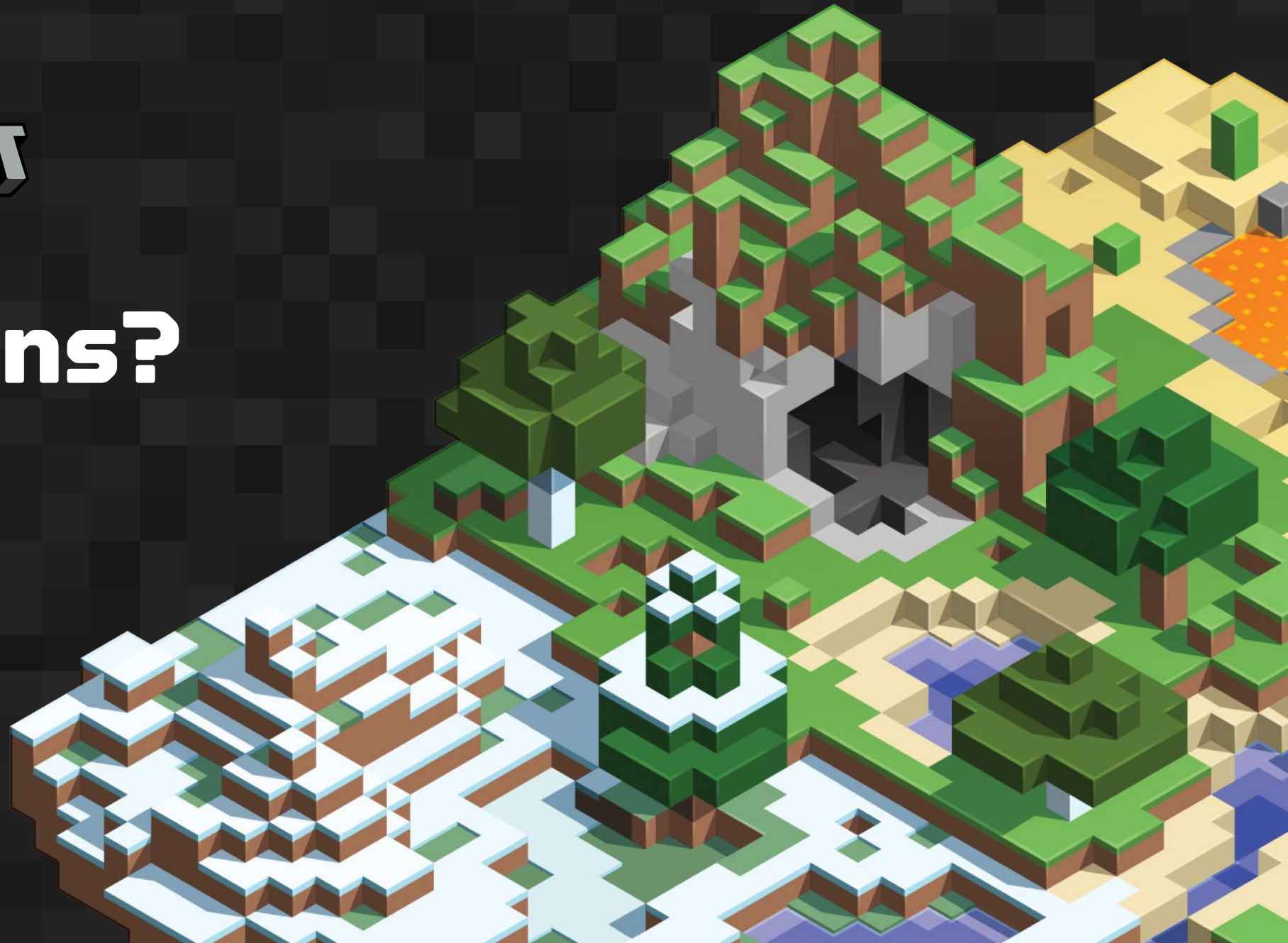
```
1 # show_title needs a title, subtitle, and action bar string
2 myvar1 = "Coding is great!"
3 myvar2 = "gotta have it"
4 myvar3 = "every day!"
5 show_title(myvar1, myvar2, myvar3 )
6 # Use variables to arrange your text for each parameter!
```

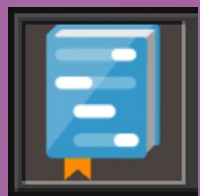
Below the code editor is a grey button with a play icon and the text "Run". To the right of the code editor is a small window showing the Minecraft game interface. The top part of the window shows the word "MINECRAFT" in red, pixelated letters. Below that is a green grassy area. At the bottom of the window, there is a chat window with the text "Billboards Let your message be heard in Minecraft!".

MINECRAFT

EDUCATION EDITION

Questions?





03 Variables

Variables

One of the most important early concepts for students to understand is the purpose and use of variables in coding.

Many students already understand the environment of Minecraft.

They understand that there are conditions in Minecraft that change, such as the weather.

Making the connection to variable use is easier because students can observe for themselves what the changing value of the variable "means" in the game.



Variables

world.weather

- In the new Notebook interface, world.weather is the name of the location which stores the current weather in their Minecraft Education game.
- In the coding window, the current state of the weather can be accessed as well as changed.
- Try this out as an exciting introduction to dealing with variables in Minecraft Python coding in Notebook.



```
# save the current weather
climate = world.weather
# echo data
say(climate)
# change the weather
world.weather= "thunder"
```

To extend this activity

Search out other values to use besides thunder. How can students research this information and test their guesses in the game? (Hint: the terms in the chat commands are frequently similar or even identical!)

Variables

Creating variables with strings for say commands

Can students improve on the greeting?

```
1  # Say Hello!
2  greeting = "Hello World!"
3  say(greeting)
4  # Introduce yourself!
5  myname = "Anonymous"
6  say(myname)
7  # try both
8  say(greeting, myname)
```

Student names are not "Anonymous" – of course!

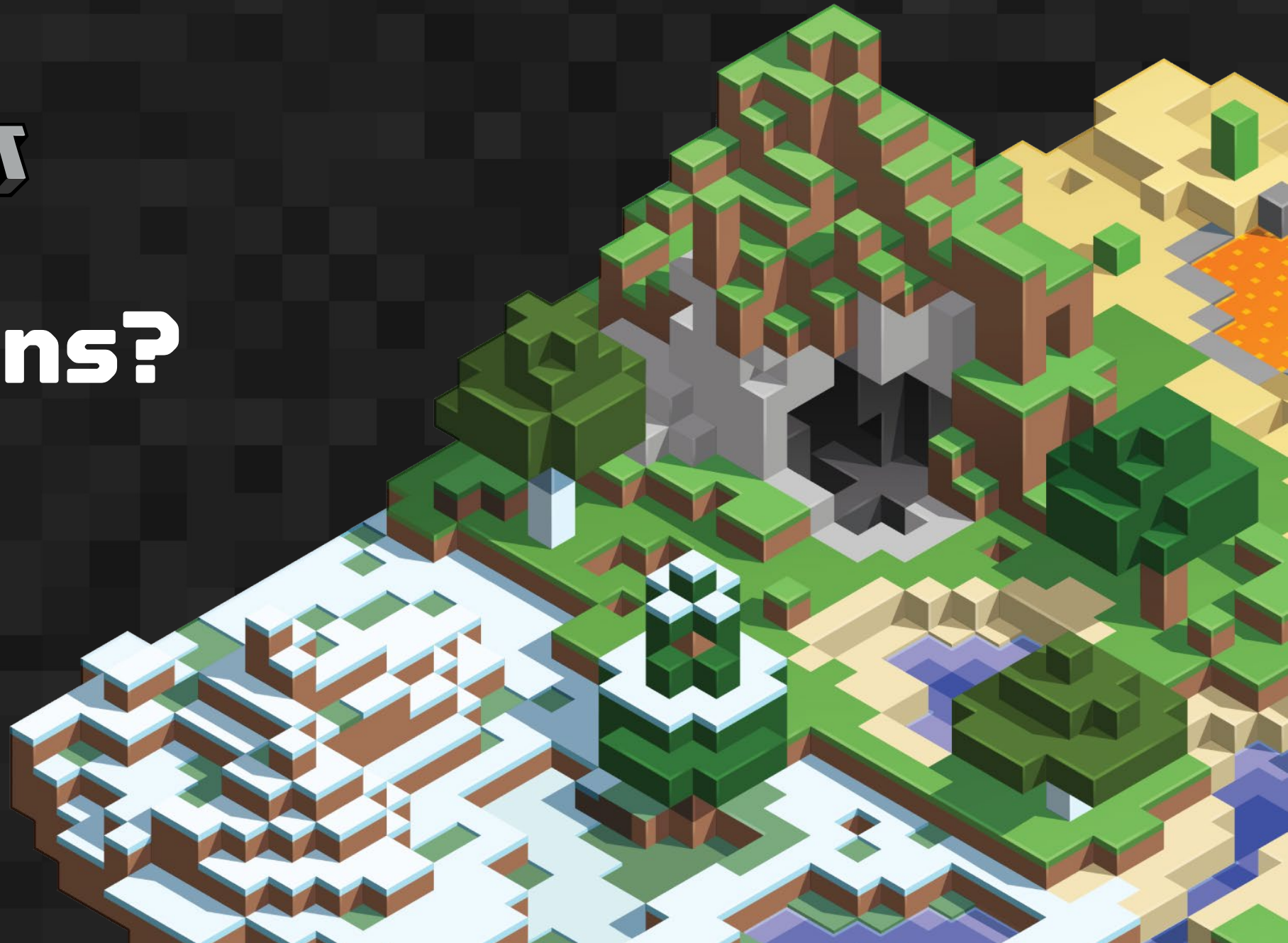
Have them edit the code to add their own names.

One trick programmers learn about strings is that adding a space at the end of a string can make the output look better when strings are **concatenated** (added together.)

MINECRAFT

EDUCATION EDITION

Questions?



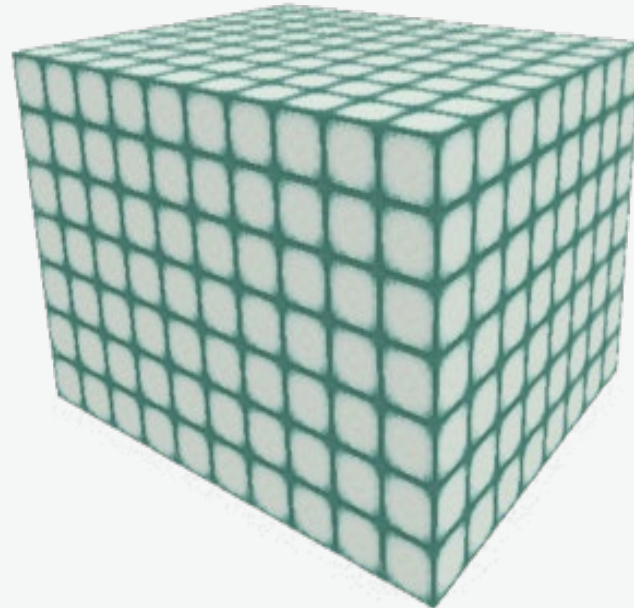


04 Coordinates, Syntax and Parameters

Coordinates, Syntax and Parameters

Coordinates are a very important kind of information in a world made of blocks! Coordinates give the exact location of blocks – and of your player and important locations - in the Minecraft world.

We will look at ways to correctly use coordinates in the Python Notebook editor, utilizing the `world.set` and `world.fill` commands.



Coordinates, Syntax and Parameters



What is Syntax?

Languages use syntax, a system of order that indicates meaning.

Syntax has to do with using symbols and words in the correct order to express meaning.

Sometimes we can understand things even though they are not in the expected order.

It can be confusing, though. This is why we tell jokes about things like commas. “Let’s eat, Grandma” is very different from “Let’s eat Grandma.”

- **Usually, the order of the words and symbols is very important to the meaning of commands.**
- **Python Notebooks has more than one way in which coordinates can be written.**
- **This can be powerful and helpful. Some commands allow you to be flexible and pick the coordinate syntax you prefer. Other commands are more picky.**

Coordinates, Syntax and Parameters

Parameters

The command **world.set** requires two parameters. Parameters give commands the information they need to do their job.

world.set needs location and a block to place in that location.

We think of the command like this:

world.set(<location>,<block>)



```
1 # Coords in world.set
2 say("Hello World")
3 # World Set Coords are strictly int with parens and commas - no tilde
4 world.set((42,4,15),"cake")
5 # ... or tilde with quotes and no commas
6 world.set("~ ~ ~","cake")
7 #hybrids in quotes
8 world.set("42 ~1 15", "cake")
9 # from_me works
10 world.set(from_me(1,3,1),"cake")
11
```

Coordinates, Syntax and Parameters

Parameters

- Note that the <block> parameter is asking for a Minecraft block like cake, and the syntax is to use the correctly spelled name (capitals matter!) and to enter the word in quotation marks.
- So a <block> might be "cake" or "dirt" or "diamond_block"
- Since you know that syntax is important, you know that you will need quotation marks around the word cake and that you cannot capitalize the word or you will get a syntax error.

```
1 # Coords in world.set
2 say("Hello World")
3 # World Set Coords are strictly int with parens and commas - no tilde
4 world.set((42,4,15),"cake")
5 # ... or tilde with quotes and no commas
6 world.set("~ ~ ~","cake")
7 #hybrids in quotes
8 world.set("42 ~1 15", "cake")
9 # from_me works
10 world.set(from_me(1,3,1),"cake")
11
```

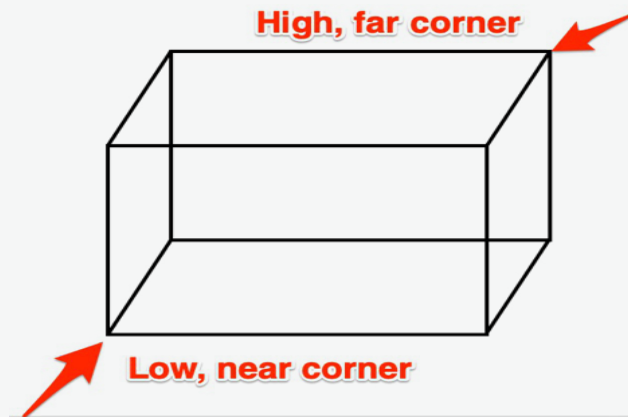
Coordinates, Syntax and Parameters

The command **world.fill** uses the coordinates of two opposite corners of a box shape to define an area to fill with a block.

We think of the command like this:

world.fill(<location> , <location>, <block>)

(To help students visualize this, grab a tissue box or cereal box and show the opposite corners.)



Coordinates, Syntax and Parameters

Blocks you may use in the set or fill commands:

"double_plant" "bed" "web" "red flower" "yellow flower" "sapling" "log" "leaves" "chest" "red mushroom" "dirt"

"stone brick" "cobblestone" "quartz_block"

"iron_block" "diamond_block" "emerald_block" "log"

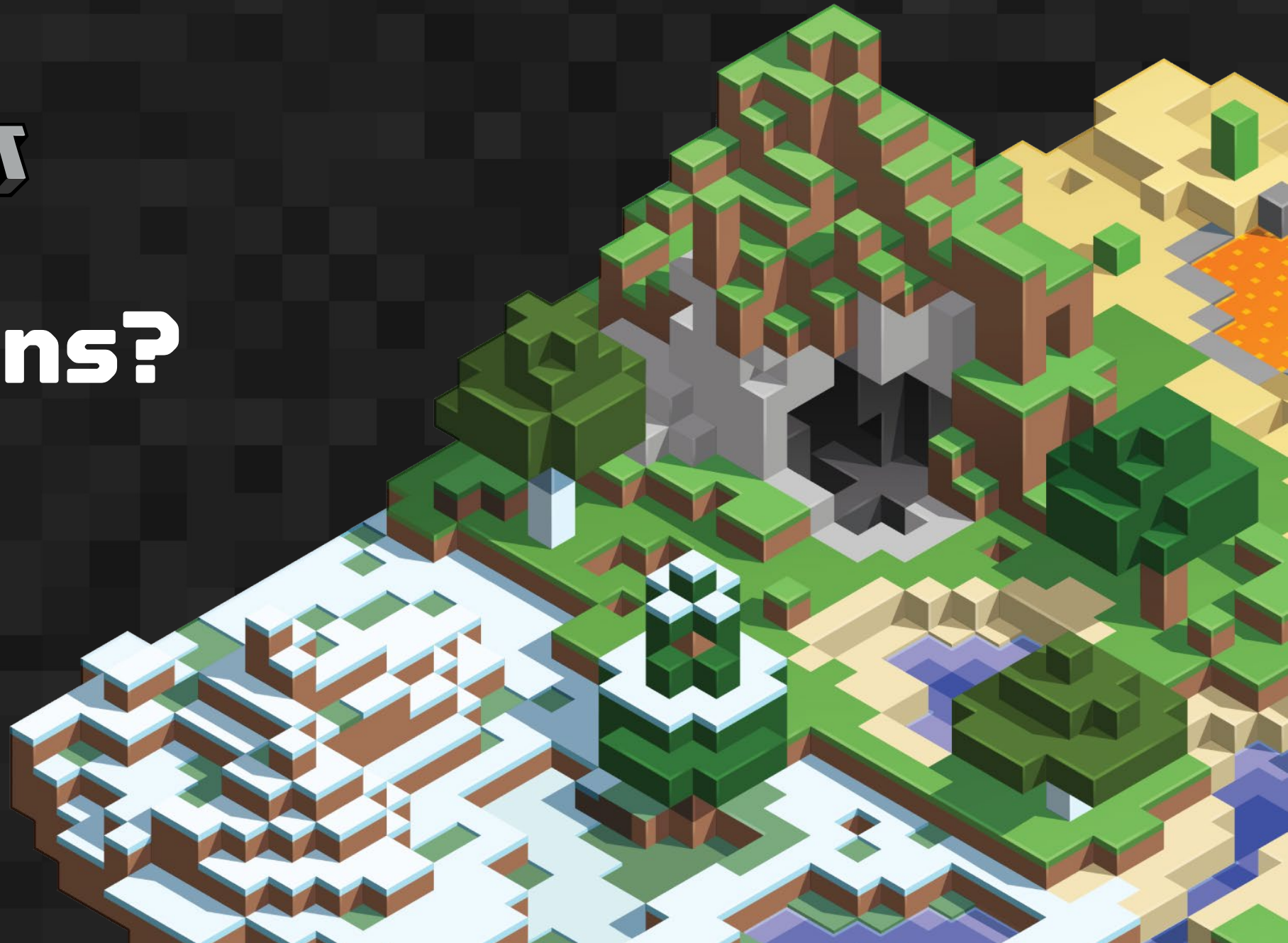
"lapis lazuli block"

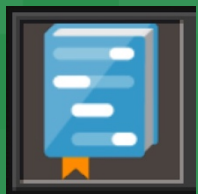


MINECRAFT

EDUCATION EDITION

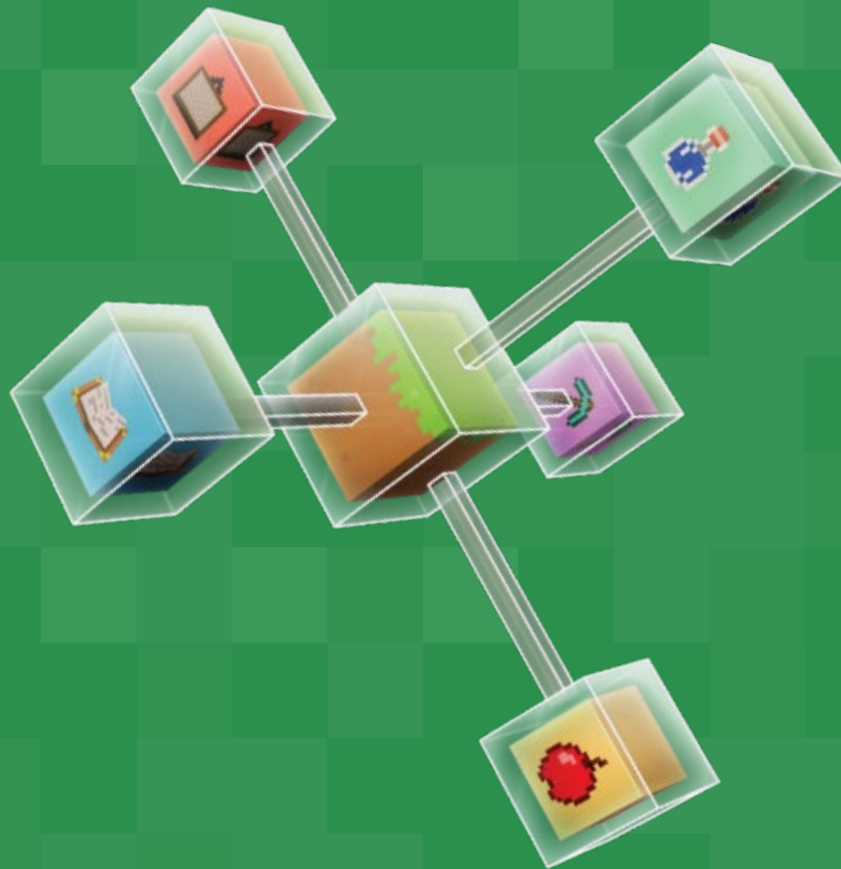
Questions?





05

Event Handlers



Event Handlers



Event handlers are the big responders in coding. They are the bridge between code and the outside world.

Event handlers are code that waits for an event, such as an action by a player in a Minecraft world, and they run code in response to this event.

Event - an action, outside the code, that is recognized and responded to inside the code.

Event handler - code which "listens for" and responds to (handles) events outside of the code.

Event Handlers

```
1 # The chat window is waiting to greet you
2 # Define a function to run when the event occurs
3 def chatter():
4     say("Pleased to Meetcha!")
5 # This event handler will continue to run,
6 # waiting for you to say, Hello
7 # in chat.
8 on_chat_event("Hello",chatter)
9
```

Examples of events in Minecraft include things like a player walking, a block breaking, or a certain word being said in chat. Code for events typically begins with the word "on"

on_chat_event(<chat command>, <function to call>) is a built-in event handler in the Notebooks editor.

Event Handlers

Here we have the on chat command waiting for "Hello" in chat, so it can call the function called chatter.



```
1 def chatter():
2     say("Pleased to Meetcha!")
3     on_chat_event("Hello",chatter)
4
```

Run Reset Code

The event handler is "listening for **Hello** in chat. When it hears Hello, it will call the function named **chatter**.

The code is entered, and and run is chosen, but no visible change will occur in the game until the event handler "hears" what it is "listening" for.


```
1 def chatter():
2     say("Pleased to Meetcha!")
3     on_chat_event("Hello",chatter)
4
```

Run Reset Code

Event Handlers

```
1 def chatter():
2     say("Pleased to Meetcha!")
3     on_chat_event("Hello",chatter)
4
```

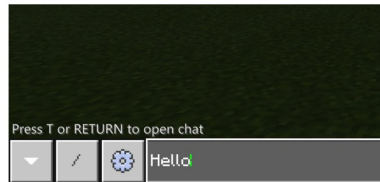
■ Stop ↻ Reset Code

 Execution is in process

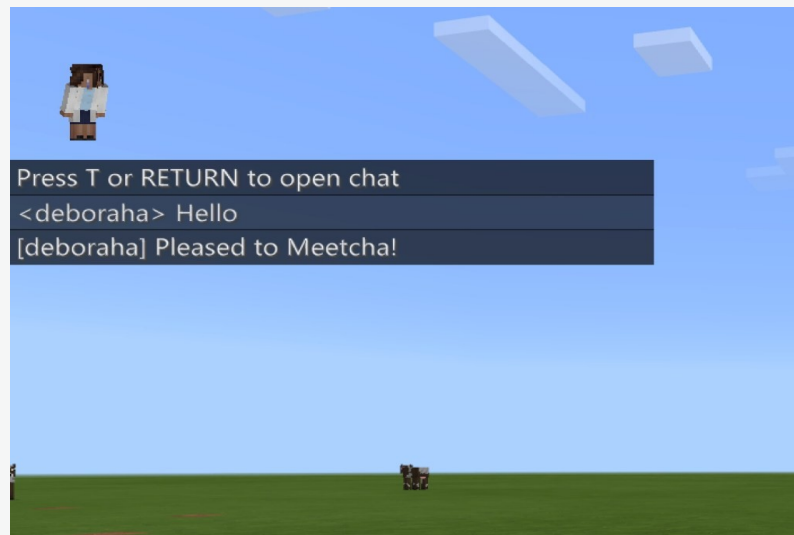
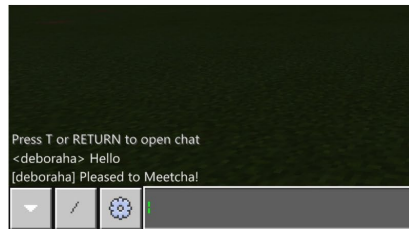
Note the "Execution in process" message at the bottom. Event handlers do not run and then stop running the way other procedures do. They are intended to continue running in the background, constantly checking to see when they should respond.

Event Handlers

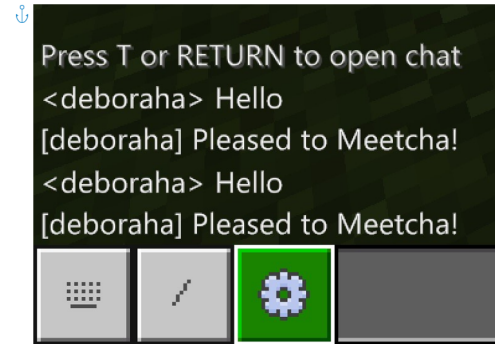
When you go back into your world, you can go to [chat](#) (type T) and enter the word Hello.



Success!

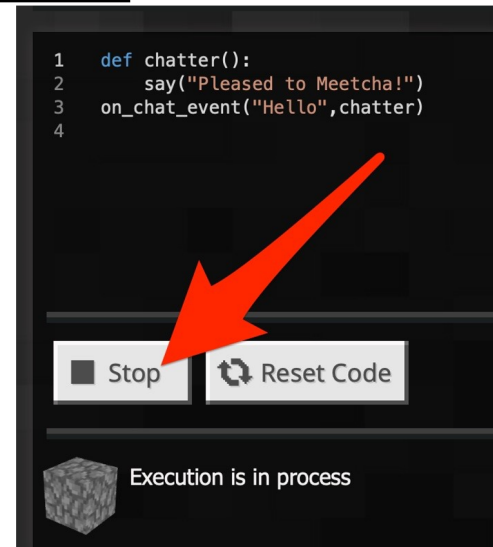


Event Handlers



Even though the code has "run" once, it isn't done. It is still executing, waiting to respond again. If you type Hello again, it will reply again. Friendly, right?

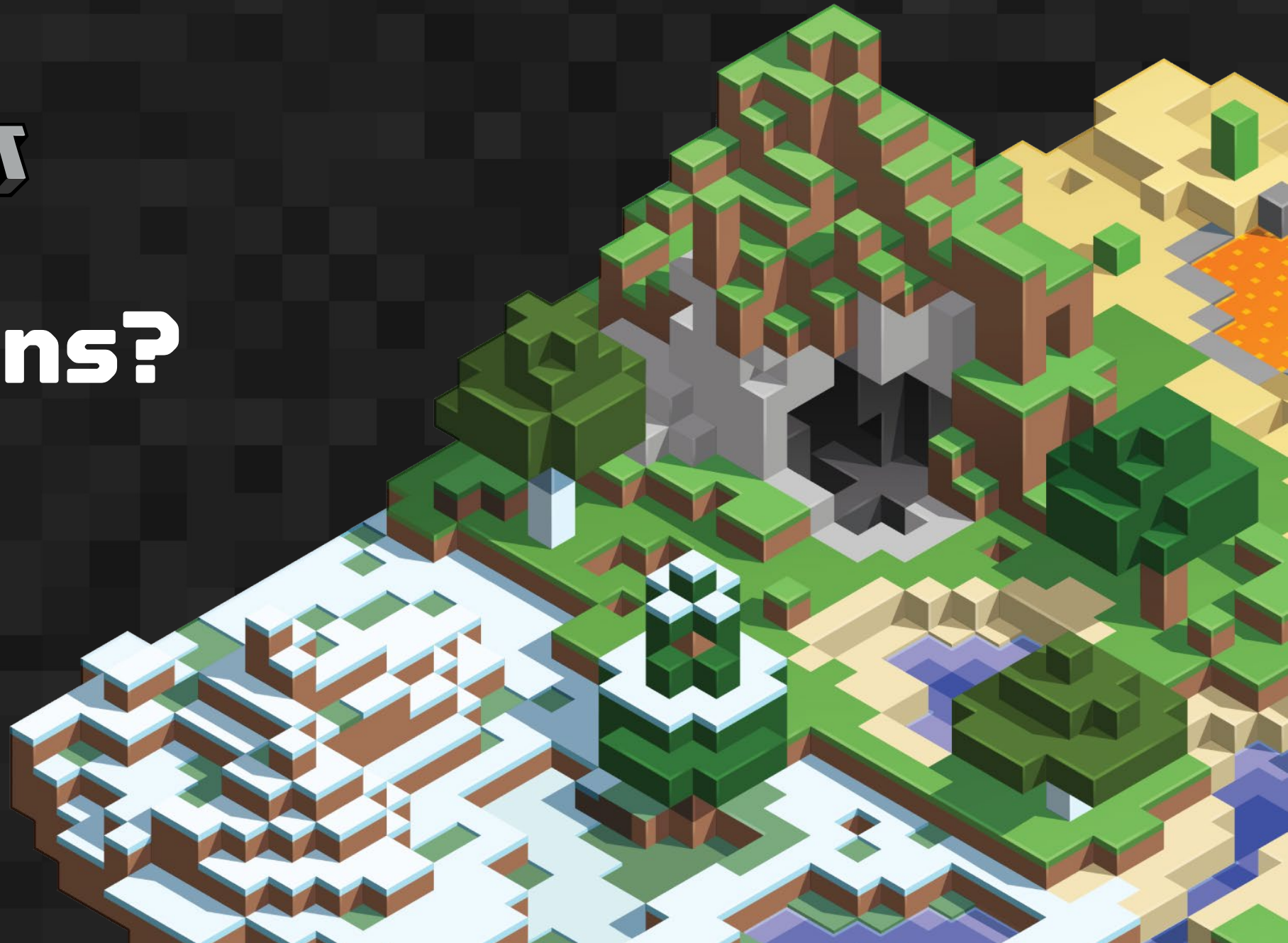
You will have to end the process using the stop button, or by exiting Minecraft if you want it to stop.



MINECRAFT

EDUCATION EDITION

Questions?



Acknowledgements

Debbie Alexander is an educator, computer scientist and Global Minecraft Mentor and a grandma! Debbie regularly contributes to the Community Boards and to this guide by providing her insights into teaching Python with Minecraft and Azure Notebooks.



MINECRAFT
EDUCATION EDITION

Thank you!

