

CODE THESE MINECRAFT MODS

Bedrock Edition



Code These Minecraft Mods: Bedrock Edition Copyright © 2022 by Neuron Fuel, Inc.

All rights reserved.

No part of this work may be reproduced or transmitted in any form or by any means without the prior written permission of the publisher.

For complete Terms of Service, please visit:

<https://www.tynker.com/terms/>

For bulk sales, licensing, or other inquiries, please contact:

info@tynker.com

Tynker and the Tynker Logo are registered trademarks of Neuron Fuel, Inc. Other product and company names mentioned herein may be the trademarks of their respective owners. Rather than use a trademark symbol with every occurrence of a trademarked name, we are using the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The information in this book is distributed on an “As Is” basis, without warranty. While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

All characters in this publication are fictitious, and any resemblance to real persons, living or dead, is purely coincidental.

CONTENTS

A Note for Parents

vi



Chapter 1

The Block Editor

1

The fundamental “unit” of Minecraft. You break blocks, craft blocks, and place blocks. By editing a block using Tynker’s Block Editor, you can change the color and texture of your world. A pixel editor.



Chapter 2

The Item Editor

7

Items are usable in-game objects. Any object in a player’s Inventory or in their hands is an item. Some familiar examples of items are Apple, Iron Pickaxe, or TNT. A pixel editor.



Chapter 3

The Skin Editor

15

Customize your player’s appearance in Minecraft by changing your Skin! A pixel-editor.



Chapter 4

The Mob Editor

23

“Mob” is Minecraft slang for animals, enemies, and anything else that moves. You can change a mob’s looks, but you can also change how the mob behaves using code blocks. Take the next step: Edit the 3D model or “skeleton” of your mob—and write an algorithm to control its in-game behavior. 3D model editor and simplified code editor.

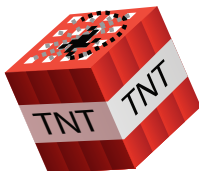


Chapter 5

Modding Basics

35

If you want to take total control of your Minecraft world, you need to learn to write mods. Full code editor.



Chapter 6

Modding Projects

53

Explore key Minecraft concepts like absolute and relative positions, as well as block data values. Learn how to teleport the player, summon mobs in a flash, and more.



Chapter 7

Minigame Mods

67

See practical applications of computer science concepts like variables, lists, and functions, as you create your own original minigames. Learn how to keep score, organize your code, and much more.



Appendix A

Where to Go From Here?

87

Looking for more Minecraft resources? Interactive courses? Find a complete course listing, and see what's compatible with Bedrock AND Java! Find it all here. A Minecraft Course Catalog.

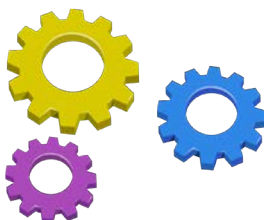


Appendix B

Glossary

93

Totally new to Minecraft? Learn some of the key terms. And look up key computer science concepts, too.



Appendix C

Troubleshooting Tips

99

Nobody said coding was easy. Learn some tips for troubleshooting common errors, learn how to ask for help, and more.

A NOTE FOR PARENTS



There's been a lot of talk in the media lately about computer programming and why it matters to your child's future.

So, why is coding so important? Of course, we rely on code in the technology we use every day—our phones, thermostats, televisions, and cars wouldn't exist without code.

But coding is also an inherently creative medium. With the ability to code, your child will be able to use their new skills to create almost anything they imagine! And getting started early means children can become fluent in code, just like they would any new language.

But... Why Minecraft?

Minecraft is the perfect visual environment to explore coding and computer science. Even without using code, Minecraft players imaginatively build anything that's in their heads. The open-ended nature of Minecraft instills a culture of creativity, with the added benefit of needing grit and problem-solving skills to bring your ideas to life.

Using Tynker, kids connect colorful blocks of code together to write programs, a little bit like LEGO bricks. And they can see the results of their work right away, acted out in the Minecraft world.

Imagine giving yourself superhero powers, like the Hulk, using code:

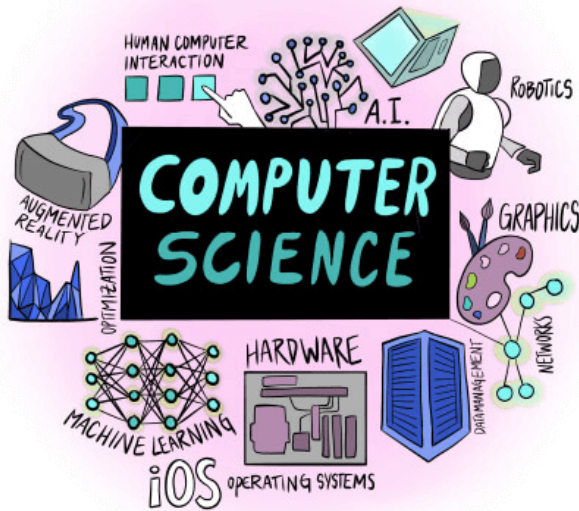


Or constructing a volcano, in a flash:



When you engage kids in activities that they're interested in and passionate about, they're incredibly motivated to problem-solve, debug, and learn.

In short, Minecraft can bring a lot of joy and familiarity to a difficult new task:



When you are making predictions, modifying other peoples' code, then writing your own custom code to achieve your particular goal in Minecraft, you're actually learning **computer science**!

What Do I Need to Run Tynker?

Tynker runs in a Web browser (like Chrome, Edge, Firefox, or Safari), with no software to install.

You need an Internet connection and a computer to run Tynker, that's it!

Tynker even runs on iPad. Install the Tynker Minecraft app called **Mod Creator** from the App Store.

Parents can check their children's progress and have multiple logins for each child. Create a free account at ***tynker.com*** now!

Which Version of Minecraft Do I Need?

Minecraft Bedrock is the most recent iteration of Minecraft, and you should use that if you are buying Minecraft for the first time. You'll be able to use all of Tynker's modding tools, as well as the latest and greatest interactive Tynker Minecraft courses. Minecraft Bedrock supports two platforms:

- **For Windows 10.** Working on a Windows PC? If you don't already have a license, you can purchase and install Minecraft at ***minecraft.net***—or from the Microsoft App Store.
- **Pocket Edition.** If you're following along on an iPad, you can also use Minecraft Pocket Edition. Install it from the Apple App Store.

You are currently reading the Bedrock edition of this book, but a Java version is also available (tynker.com/ebooks).





Warning

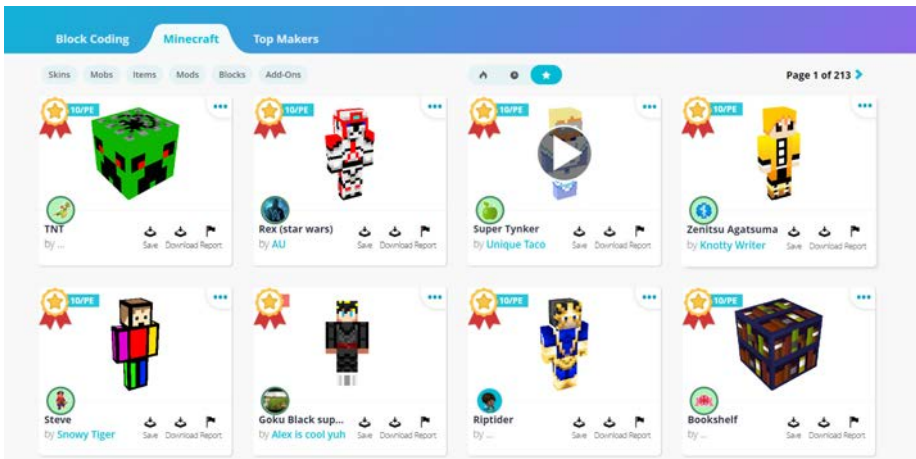
This book is designed to be used with Minecraft Bedrock *only*. However, if you already have **Minecraft Java** (sometimes called Minecraft Classic), don't worry: You don't need to buy Minecraft again. All of the Tynker modding tools are compatible with Minecraft Java—several courses are available for Minecraft Java as well.

The Tynker Community

What happens when more than 60 million kids share coding projects with one another in a collaborative space? A whole lot of fun—and loads of teaching, sharing, and support!

Your kids can share their work and get feedback or find inspiration in thousands of our public projects:

<https://www.tynker.com/dashboard/student/#/community>



The community is constantly changing and improving, but one thing stays the same: it's always inspiring!

Tynker for Parents

Learn about different home subscriptions, coding paths, and tools to ensure your child is making the most of your Tynker subscription. View your child's course progression, assign courses, or just let them explore!

[*tynker.com/parents*](https://tynker.com/parents)

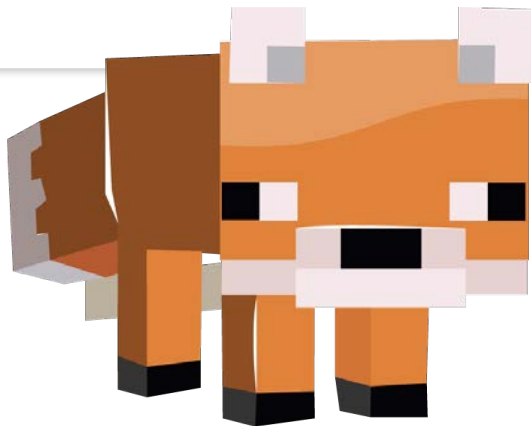
Tynker for Teachers

Tynker has a lot to offer teachers:

- Dozens of Hour of Code activities
- Free weekly projects
- Free professional development and training sessions
- Courses designed for all grade-levels

For a complete course catalog and more information, please visit:

[*tynker.com/school*](https://tynker.com/school)





Chapter 1

The Block Editor

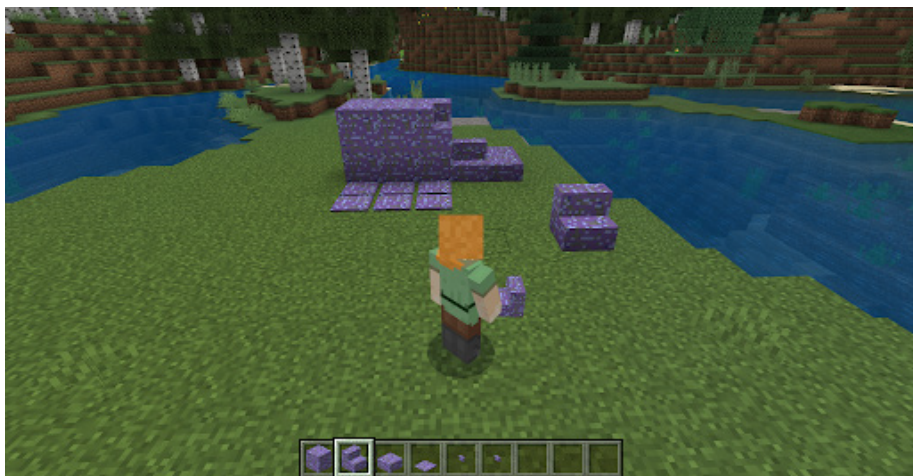
The fundamental “unit” of Minecraft. You break blocks, craft blocks, and place blocks. By editing a block using Tynker’s Block Editor, you can change the color and texture of your world.

Start at tynker.com/modbedrock



PROJECT IDEAS

Wondering what you can do with the Blocks Editor? Here are a couple ideas to get you started.



Create a house in **your** favorite color! In this example, boring **gray** stone has been edited to become **PURPLE**. Notice how all the items made of stone blocks become purple, automatically! These color changes affect **all** stone materials in this world!

You can even add your favorite **pixel art** to a block, so you can hang it up like painting in your Minecraft house. A Minecraft block is 16×16 pixels wide, so you have to be efficient when you draw!

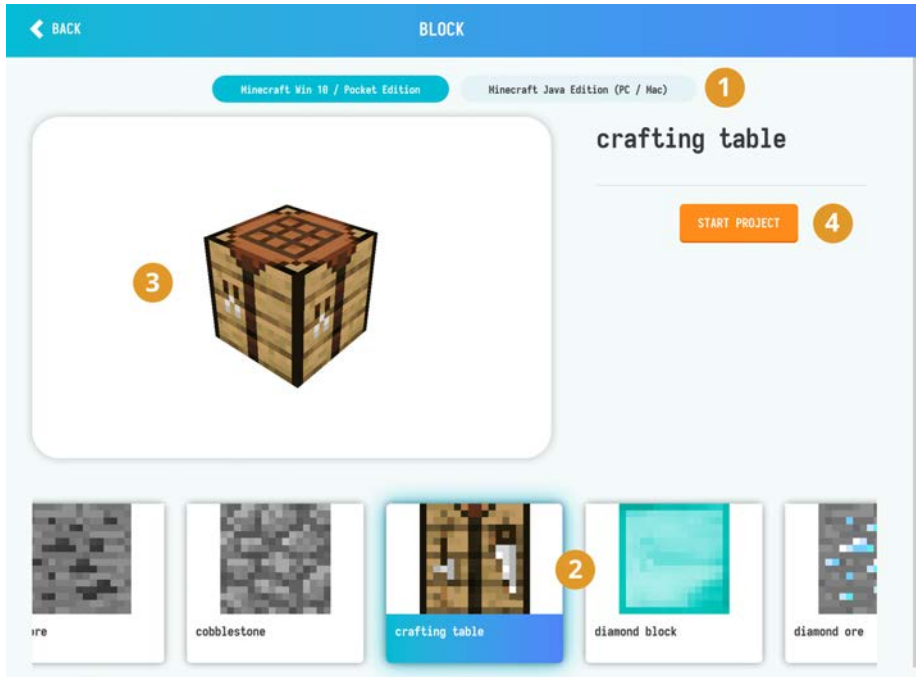




LAUNCH THE BLOCK EDITOR

Login to your Tynker account at **tynker.com** or start the app.

From your Dashboard, click on the **+Create Project** button, then find the **Block Editor** in the **Minecraft** category. You can choose your target block here.

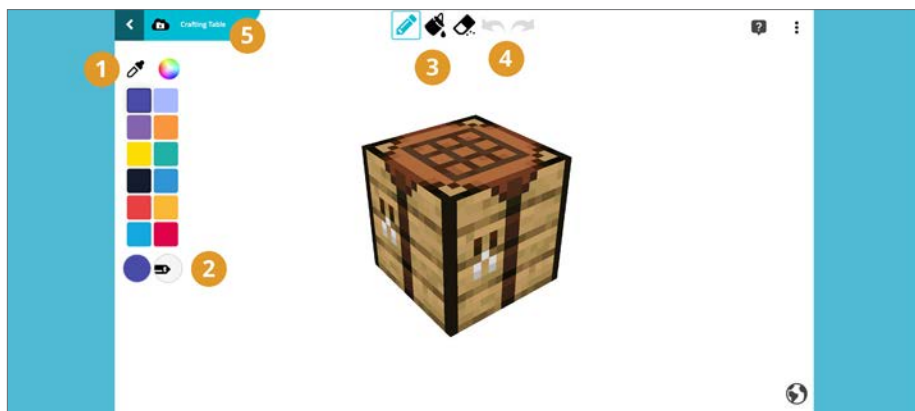


- 1 First: Choose your **Version** of Minecraft
- 2 Scroll through available Blocks in the horizontal scrollbar
- 3 See a **Preview** of your Block here
- 4 Click **Start Project** to begin editing your Block

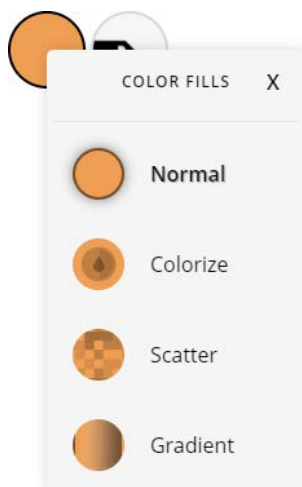


THE BLOCK EDITOR

Let's take a tour of what you can do with the Block Editor.



1 Color Selection Tools. Choose one of the options here, or use the **Eyedropper** to choose a custom color used in the project. Click on the **Rainbow Circle** icon to choose a totally custom color of your choice. If you click on the active color, you can change the current Color Fill style, which allows you to paint using variants of similar colors. Give it a try!

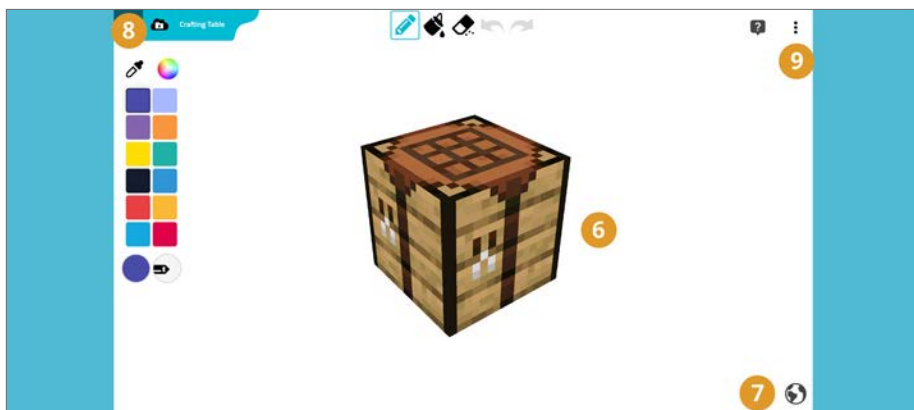


2 Brush Selection. Displays your currently selected color and allows you to change the brush size.

3 Tool Selection. Choose between Pencil, Fill Tool, and the Eraser here.

4 Undo and Redo. Make a mistake? Click on these buttons to undo or redo actions.

5 Skin Name. Give your block a unique name here.



6 3D View. Click and drag the canvas to spin your block in the 3D space. You can see how the block looks from every angle.

7 Block Preview. See how your block looks in a 3D world. Click the globe to return to editing.

8 Download Block. All done with your customization? Download your Block here and remember where you save the file. Follow the instructions to deploy your Block to a Minecraft world.

9 Switch Block. Change your mind? Use the Three Dot Menu ⋮ to choose another starter block.

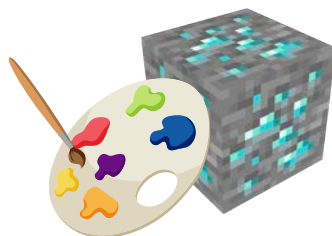
IMPORT YOUR BLOCK

So you've done it, you've completed your first block design.

Download the file—and follow the onscreen instructions to load your addon into Minecraft!

Here's the big idea:

1. Paint your block in Tynker



2. Download your work as an **.mcaddon** file, and remember where you save your file. Open up the file using Minecraft (double-clicking it is easiest). This add-on file is also called a **Resource Pack**, and it contains the artwork you just created.

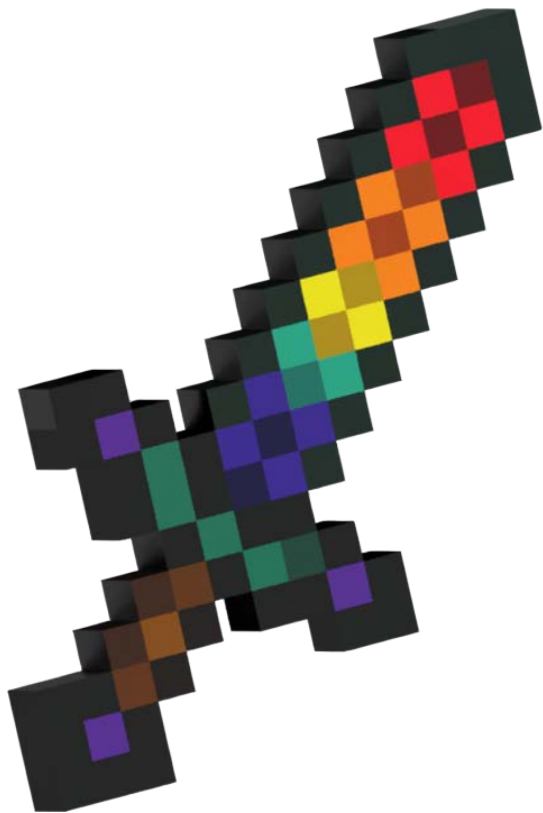


Note that iPad users who have “Allowed Access” to Tynker’s Mod Creator app will have the resource pack automatically added to Minecraft.

3. Create a new world in Minecraft, then enable the Resource Pack you just created. From the Create World screen, go to **Resource Pack, My Packs**, then select your block. It will then appear in your **Active** Resource Packs, like so:



4. You should see the block right away in your Minecraft world if it's a common one. You can also enable Creative Mode to give yourself ALL possible blocks right away to see your new, cool, custom block! Switch modes by using the **/gamemode creative** command in chat.



Chapter 2

The Item Editor

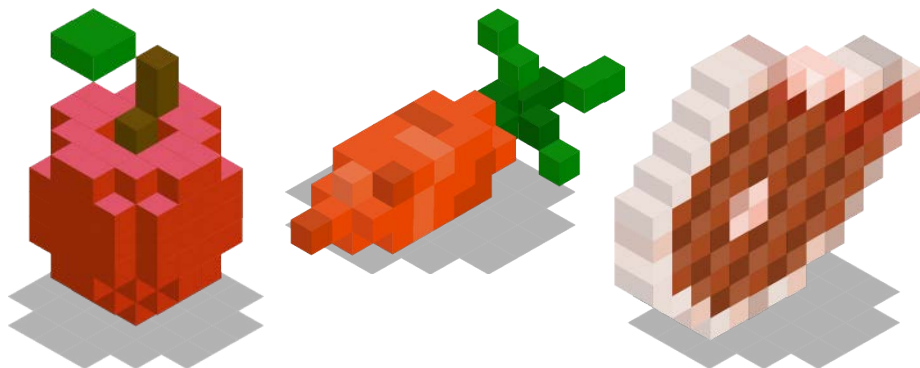
Items are usable in-game objects. Any object in a player's Inventory or in their hands is an item. Some familiar examples of items are Apple, Iron Pickaxe, or TNT.

Start at tynker.com/modbedrock

Weapons, tools, and equipment are items too, like these:

- Diamond Armor
- Stone Hoe
- Fishing Rod

Using the **Tynker Minecraft Items Editor**, you can customize **how an item looks**.



i Hint

Want to change how an item behaves, not just how it looks? Use the **Mod Editor** and write custom code for your item. For example, if you want to make an enchanted item, use the **Mods Editor** instead of the **Item Editor**. Skip ahead to read **Modding Basics: Chapter 5**



PROJECT IDEAS

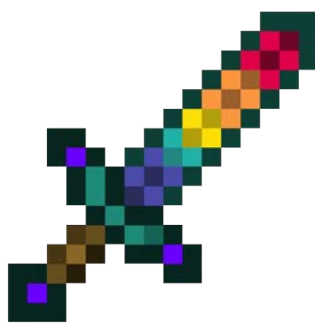
Wondering what kinds of projects you might create with the Item Editor? Here are a few ideas.



Make your TNT blue! Or change the color of any item you like.



Make a custom sign!



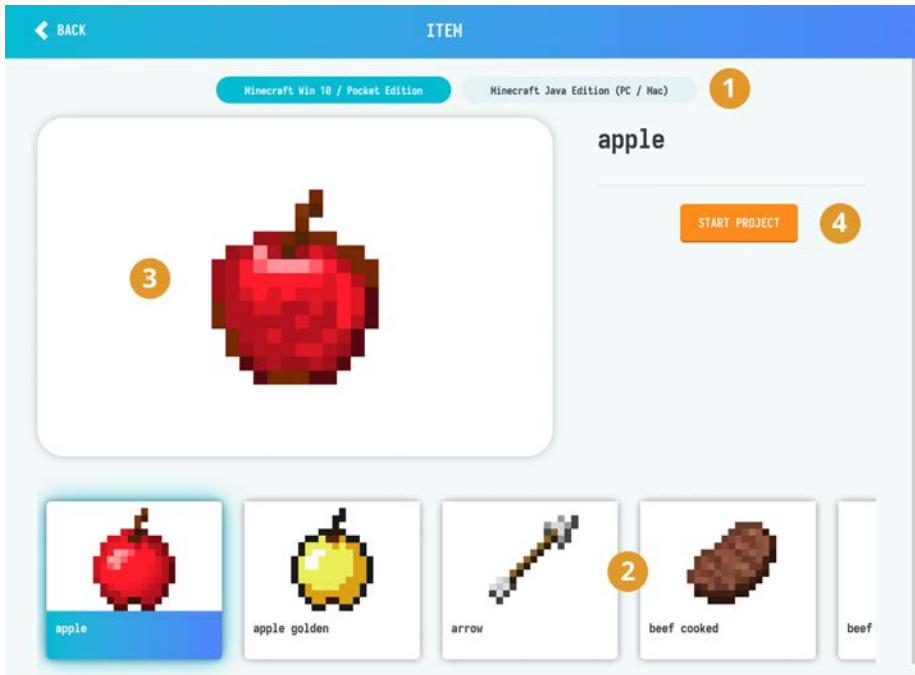
This diamond sword has the power of rainbows.



LAUNCH THE ITEMS EDITOR

Login to your Tynker account at **tynker.com** or start the app.

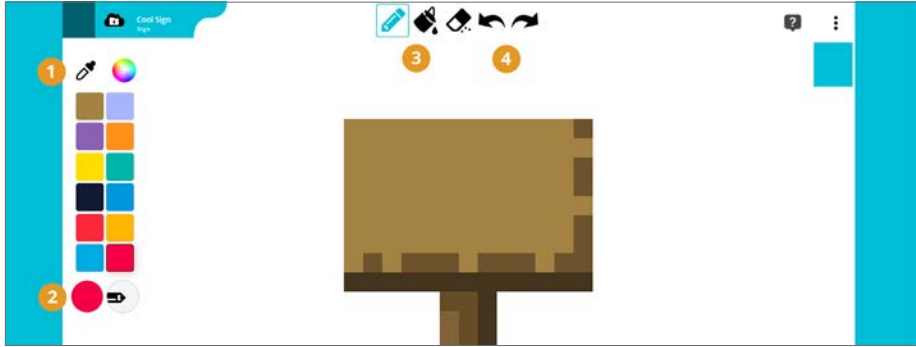
From your Dashboard, click on the **+Create Project** button, then find Items Editor in the **Minecraft** category. You can choose your starting item here.



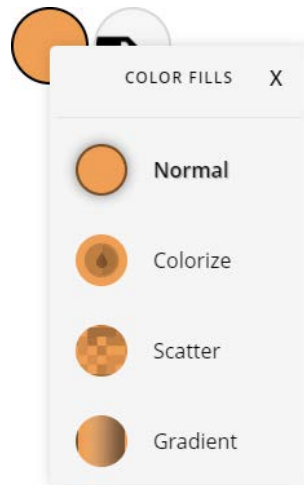
- 1 First: Choose your **Version** of Minecraft
- 2 Scroll through available items in the horizontal scrollbar
- 3 See a **Preview** of your item here
- 4 Click **Start Project** to begin editing your item

THE ITEMS EDITOR

Let's take a tour of what you can do with the Item Editor. In this example, we've started by opening the **Sign** item, then erased the text on the sign to write our own custom message:



1 Color Selection Tools. Choose one of the options here, or use the **Eyedropper** to choose a custom color used in the project. Click on the **Rainbow Circle** icon to choose a totally custom color of your choice. If you click on the active color, you can change the current Color Fill style, which allows you to paint using variants of similar colors. Give it a try!



2 Brush Selection. Displays your currently selected color and allows you to change the brush size.

3 Tool Selection. Choose between Paintbrush, Fill Tool, and the Eraser here.

4 Undo and Redo. Make a mistake? Click on these buttons to undo or redo actions.



5 Name and Item Type. Give your item a unique name here. Notice how the type of item is displayed below it.

6 Download Item. All done with your customization? Download your Item here and remember where you save the file. Follow the instructions to deploy your item to a Minecraft world.

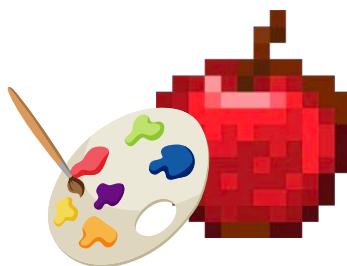
7 Switch Item. Change your mind? Use the Three Dot Menu  to choose another Item.

DEPLOY YOUR ADD-ON

So you've done it, you've completed your first item design. Download the file—and follow the onscreen instructions to load your item into Minecraft!

Here's the big idea:

1. Paint your item in Tynker



2. Download your work as an **.mcaddon** file, and remember where you save your file. Open up the file using Minecraft (double-clicking it is easiest). This add-on file is also called a **Resource Pack**, and it contains the artwork you just created.



(Note that iPad users who have “Allowed Access” to Tynker’s Mod Creator app will have the resource pack automatically added to Minecraft.)

3. Create a new world in Minecraft, then enable the Resource Pack you just created. From the Create World screen, go to **Resource Pack, My Packs**, then select your item. It will then appear in your **Active** Resource Packs, like so:



4. You can enable Creative Mode to give yourself ALL possible items right away to see your new, cool, custom item! Switch modes by using the **/gamemode creative** command in chat.



ITEMS BASICS IN MINECRAFT

Are you totally new to Minecraft? Here are some basics about items. When you have an item equipped in Minecraft, you can usually use it in one of two ways:

- Left click (Button 1): The primary action:
 - Attack
 - Break blocks
- Right click (Button 2): Alternate actions like:
 - Placing a block
 - Shearing sheep, making farmland with a hoe
 - Eating food or drinking a potion
 - Shooting a bow & arrow (or throwing a snowball) and similar

The item hotbar at the bottom of the screen is a list of easily accessible items in your HUD (the onscreen display). Open your inventory (press **E** on the desktop version of Minecraft) and place items there. Then you can switch between your hotbar items using:

- The mouse scroll wheel
- Pressing number keys corresponding with by slot number (1, 2, 3, etc.)



Want to change these controls? They are customizable in your **Minecraft Options**.



Chapter 3

The Skin Editor

Customize your player's appearance in Minecraft by changing your Skin!

Start at tynker.com/modbedrock



PROJECT IDEAS

Wondering what you can do with the Skins Editor? Here are a few ideas to get you started.



Make a superhero



Minecraft at the beach



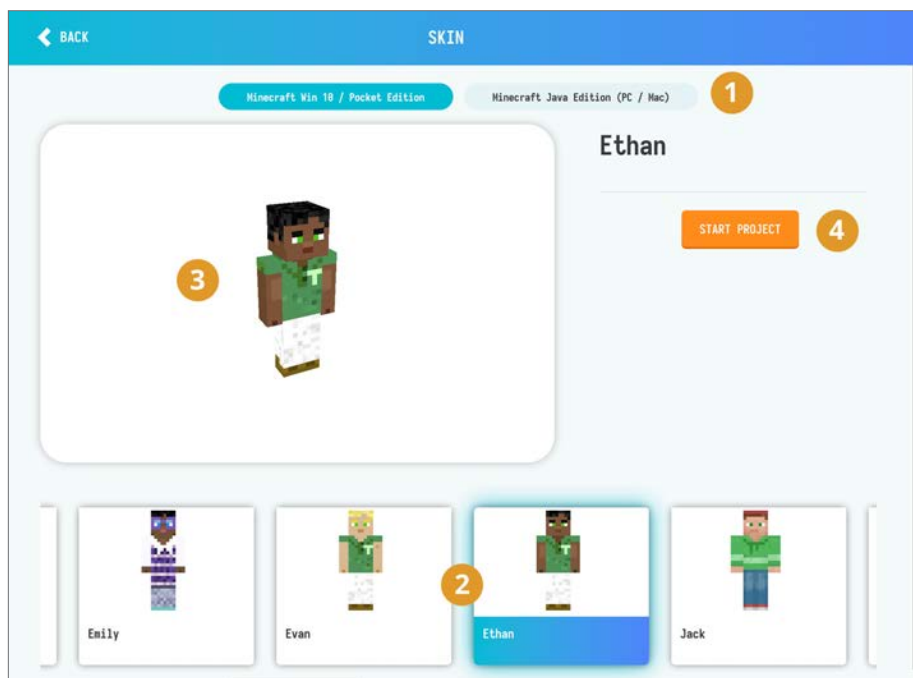
9-to-5 office worker



LAUNCH THE SKINS EDITOR

Login to your Tynker account at **tynker.com** or start the app.

From your Dashboard, click on the **+Create Project** button, then find the Skins Editor in the **Minecraft** category. You can choose your starting skin here.

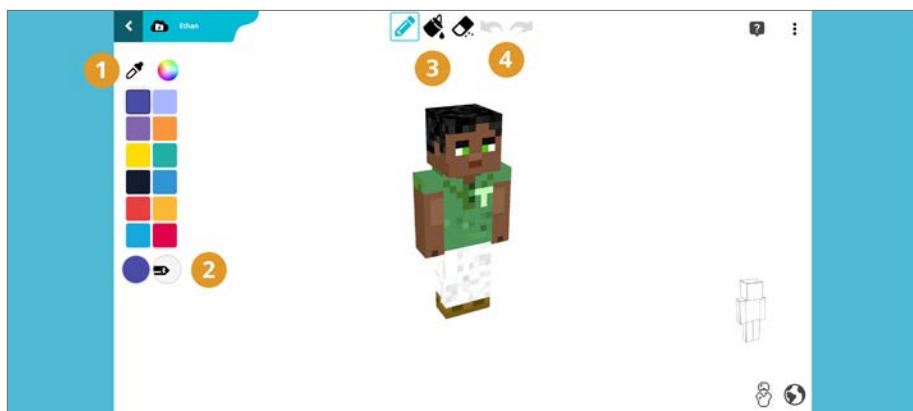


- 1 First: Choose your **Version** of Minecraft
- 2 Scroll through available starter Skins in the horizontal scrollbar
- 3 See a **Preview** of your skin here
- 4 Click **Start Project** to begin editing your Skin

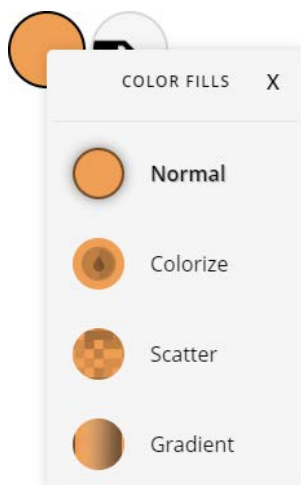


THE SKINS EDITOR

Let's take a tour of what you can do with the Skins Editor.



1 Color Selection Tools. Choose one of the options here, or use the **Eyedropper** to choose a custom color used in the project. Click on the **Rainbow Circle** icon to choose a totally custom color of your choice. If you click on the active color, you can change the current Color Fill style, which allows you to paint using variants of similar colors. Give it a try!



2 Brush Selection. Displays your currently selected color and allows you to change the brush size.

3 Tool Selection. Choose between Paintbrush, Fill Tool, and the Eraser here.

4 Undo and Redo. Make a mistake? Click on these buttons to undo or redo actions.



5 Skin Name. Give your skin a unique name here.

6 3D View. Click and drag the canvas to spin your skin in the 3D space. Don't forget to paint the back of your skin!



7 Globe: Preview in-World. See how your skin looks in a 3D world. Click the globe to return to editing. **Add Armor:** Change the body type—grow your character so they're bulkier, like they're wearing armor

8 Download Skin. All done with your customization? Download your Item here and remember where you save the file. Follow the instructions to deploy your skin to a Minecraft world.

9 Switch Skin. Change your mind? Use the Three Dot Menu ⋮ to choose another starter skin.

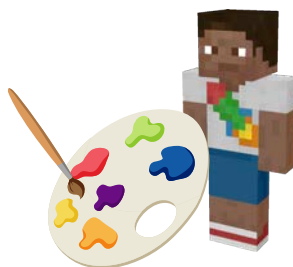


IMPORT YOUR SKIN

So you've done it, you've completed your first block design. You want to deploy it to Minecraft!

Here's the big idea of how it works:

1. Paint your skin in Tynker



2. When you're finished, download your skin as an .PNG file, and remember where you save your file



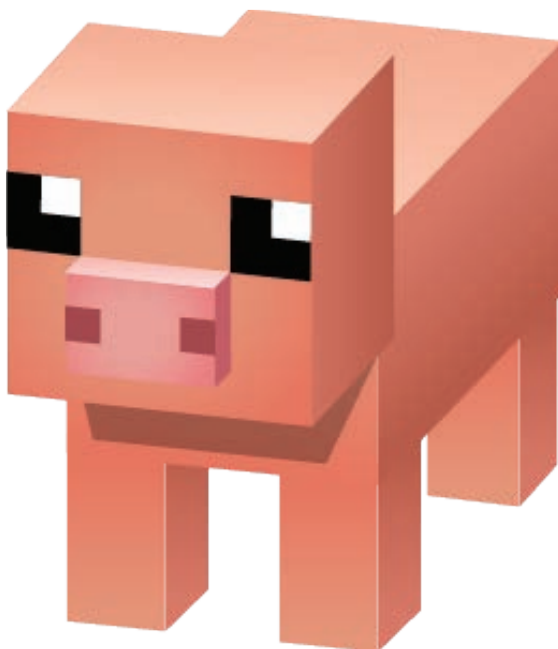
3. Then launch Minecraft. Click on **Dressing Room** or **My Profile** from the Minecraft start screen. Navigate to the Classic Skins tab, choose **Owned > Import > Choose New Skin**, then select your file.





Ready for a fun, interactive Minecraft course, designed to teach beginners from the ground up? Skip ahead to **Training Day**, and learn all the basics.





Chapter 4

The Mob Editor

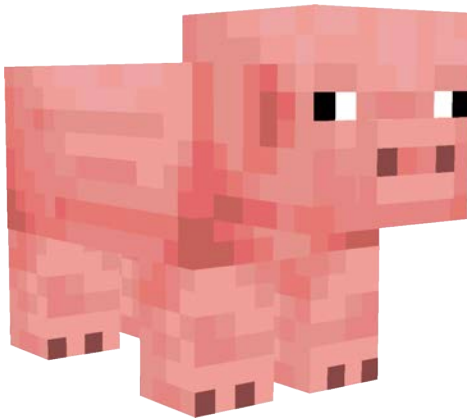
“Mob” is Minecraft slang for animals, enemies, and anything else that moves. You can change a mob’s looks, but you can also change how the mob behaves using code blocks. Take the next step: Edit the 3D model or “skeleton” of your mob—and write an algorithm to control its in-game behavior.

Start at tynker.com/modbedrock

Here are some more ideas to get you thinking. What do you want to make?

You can make it happen with the **Mob Editor**. In other words, the Mob Editor allows for game changes that are **more** than just skin-deep. Of course, you can change how an animal looks... But also how the animal **behaves**, using code blocks.

Invent whatever kind of creature you like—changing attributes, behaviors, and more, using simple scripts like this:

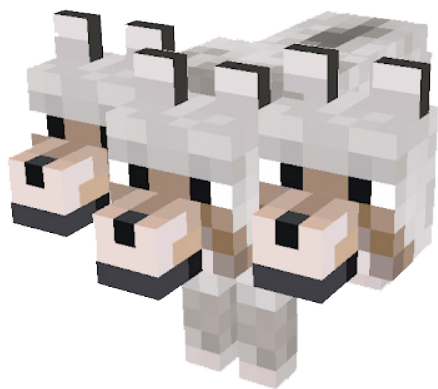




PROJECT IDEAS



This adorable crab started as a scary cave spider.



This three-headed Cerberus mob started as a wolf.



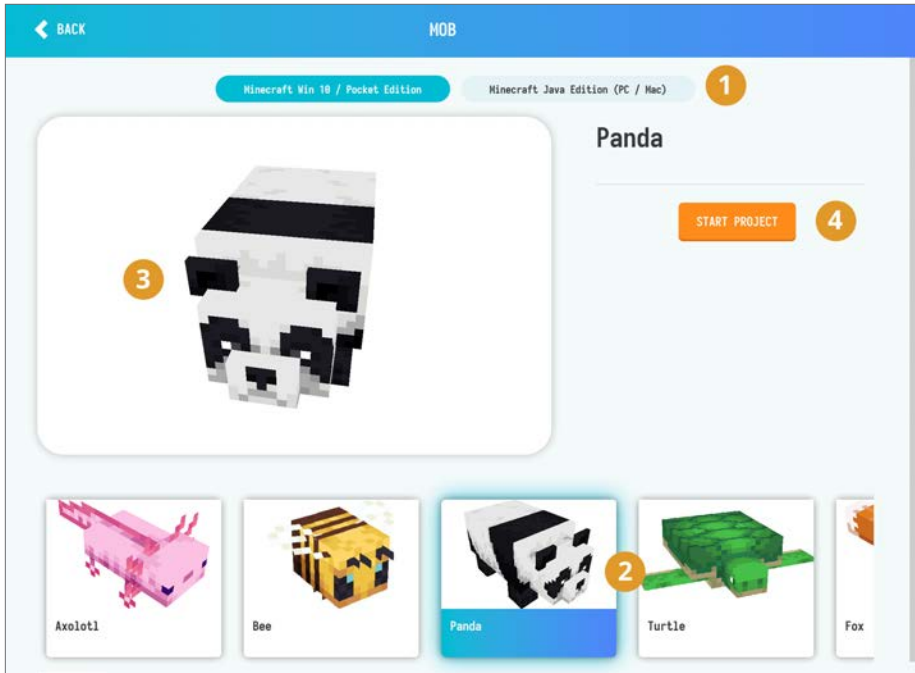
This vampire started as a bat.



LAUNCH THE MOBS EDITOR

Login to your Tynker account at **tynker.com** or start the app.

From your Dashboard, click on the **+Create Project** button, then find Mob Editor in the **Minecraft** category. You can choose your starting mob here.



- 1 First: Choose your **Version** of Minecraft
- 2 Scroll through available Mobs in the horizontal scrollbar
- 3 See a **Preview** of your Mob here
- 4 Click **Start Project** to begin editing your Mob

Hint

The starting Mob you choose matters. Choose a **similar shape** and **body type** to what you have in mind, if one is available. This will ensure that the new mob animates well within the Minecraft world. (If you make major changes to the structure of the starting model, you will likely see unexpected mob animations.)



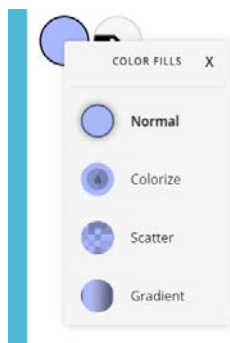
THE MOB EDITOR

Let's take a tour of what you can do with the Mob Editor.



1 Color Selection Tools. Choose one of the options here, or use the **Eyedropper** to choose a custom color used in the project. Click on the **Rainbow Circle** icon to choose a totally custom color of your choice.

Click on the color icon itself, and you'll see this dropdown menu appear. Try it out to see how it works! **Colorize** mode is a great way to switch the coloring of a mob (for example, from green to red), while keeping its textures otherwise intact.



- 2 Brush Selection.** Displays your currently selected color and allows you to change the brush size.
- 3 Tool Selection.** Choose between Pencil, Fill Tool, and the Eraser here.
- 4 Undo and Redo.** Make a mistake? Click on these buttons to undo or redo actions.
- 5 Mob Name.** Give your Mob a unique name here.



6 Click and Drag. Click and drag the canvas to spin your Mob in the 3D space. Don't forget to paint the back of your Mob, too!



7 Enter 3D Editing Mode. Click this block to enter 3D-editing mode. Flip ahead to the next section for model-editing tips and the tools available to **change a mob's shape**.

8 Add New Mob Code. Write your own custom scripts to change mob behavior. Flip ahead to the section on Coding for advice and some short scripts to try.

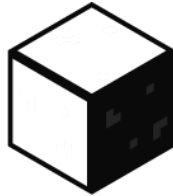
9 Block Preview. See how your block looks in a 3D world. Click the globe to return to editing.

10 Download Mob. All done with your customization? Download your Mob here and remember where you save the file. Follow the instructions to deploy your Mob to a Minecraft world.



DIGITAL SCULPTING

Now let's take a closer look at the way that models are built. Enter the 3D view of your mob by clicking this button at the top of the screen.



Try rotating the canvas to see all the different parts.

Notice how the painting tools are gone. The new tools at left allow you to change an element of your model:



Move



Scale (Grow & Shrink)



Rotate (Spin)



Delete



Duplicate (Make a Copy)

Get fine-grained control by using the 3D model shown in lower right.

Notice how slider controls appear at right for **Move**, **Scale**, and **Rotate**. Try playing with these values to get the hang of things:

X	-18.11
Y	-44.06
Z	-36.67

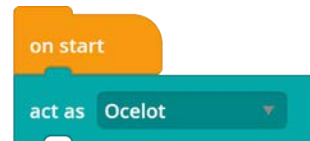
Don't forget: You can press the **Undo** (↶) button to revert a change you don't like.

MOB CODING TIPS

1 Choose Starting Behavior

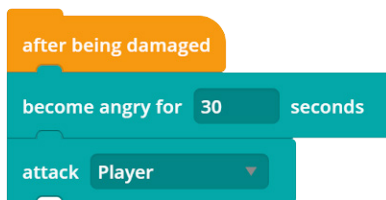
If there's already one mob that you know you want to mimic, you can choose its starting behavior to give yourself a leg-up. Some mobs are meek, while others aggressively hunt down the player.

Choose the right mob to **"act as"**.
For example, the Ocelot is shy—it runs from the player when seen.
If you know you want that kind of behavior, you can use:



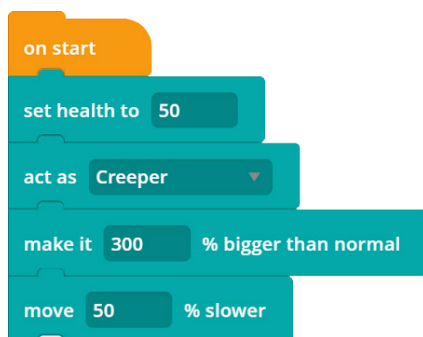
2 Events

You can trigger the mob to respond to certain events, such as “getting aggro” when damaged by the player. Here’s one idea:



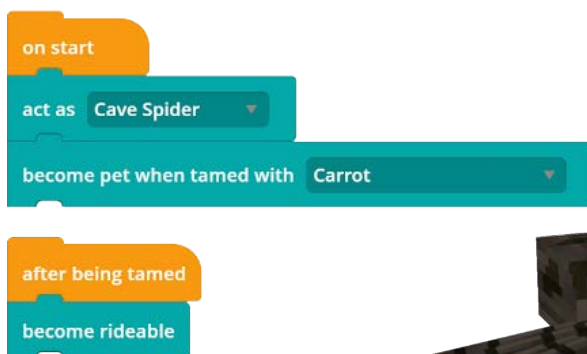
3 Mob Attributes

Wondering how to make that gigantic Creeper? Here’s an idea:



4 Tame and Ride

Want to ride your spider into the sunset? With these two short scripts, you can feed Cave Spiders carrots and they will become **tame**, then **rideable**. Giddyup.



5 Search for Blocks

Looking for a particular block or attribute? Use the search  to find the blocks you need, fast:

CODE

health

ALL

MACROS

EVENTS

set health

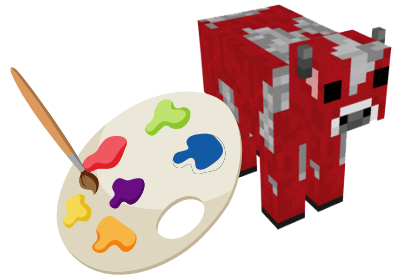


DEPLOY YOUR MOB

So you've done it, you've completed your first Mob design.

Here's the big idea on how to see your mod in Minecraft:

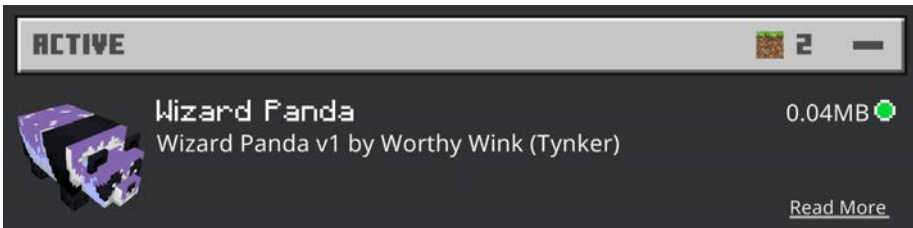
1. Paint and sculpt your mob in Tynker; Update your mob's code using Tynker, too.



2. When your design is complete, download it as an **.mcaddon** file, and remember where you save your file.



3. Open your file in Minecraft (double-clicking is simplest). Then **Create** a new world. Choose **Behavior Packs**, then **My Packs**. Enable your mod, and you should see it appear in **Active** Behavior Packs. Then go to **Resource Packs**, and again select **My Packs**. Activate your mob here in this location as well. Activating both the Resource and Behavior Pack ensures your custom mob both looks and behaves differently.



4. Load the Minecraft world. You may need to summon your mob. Try using the command in chat **/summon panda**, replacing “panda” with whichever base mob you used.





Chapter 5

Modding Basics

If you want to take total control of your Minecraft world, you need to learn to write mods.

Start at tynker.com/modbedrock

“Modding” can refer to extensions of any game, though, not just Minecraft—writing your own custom game mods is a time-tested way to learn about programming. Many young programmers got their start modding **Doom** way back in the 1990s, and there’s an enthusiastic modding scene for many popular retro and modern video games, too.

But the creators of Minecraft have taken community modding to a whole new level: It’s simple and powerful with Tynker. Let’s explore.

PROJECT IDEAS

Wondering why you might code your own Minecraft mod? You can do stuff like this:



Give yourself gear or blocks—and player effects, like Slow Falling or a Super Jump ability



Construct stuff incredibly fast! Make an golden railway as long as you like, and travel in comfort and style



Just make things weird. Maybe just make it rain cake...?



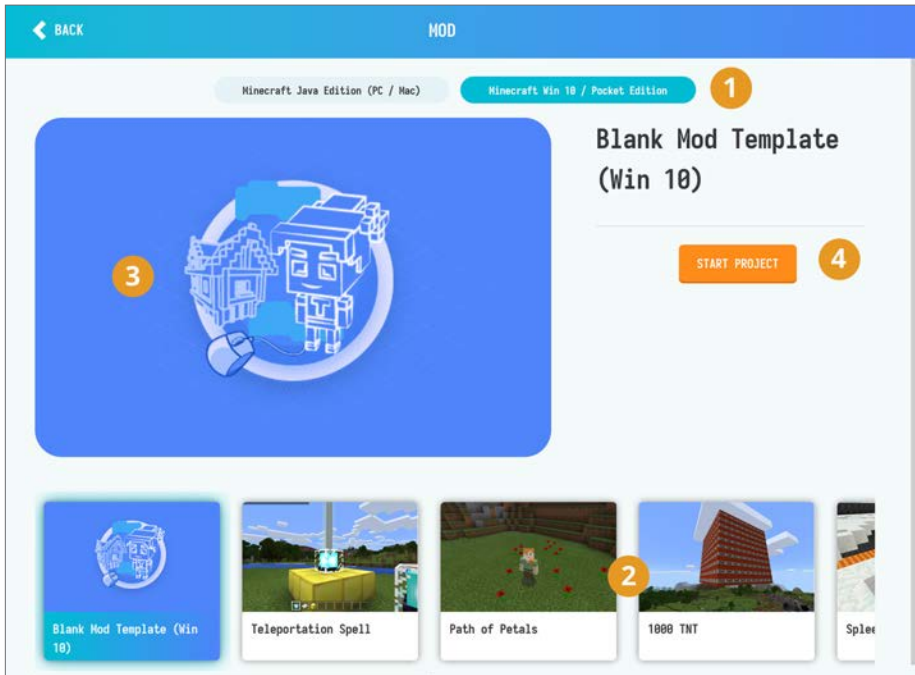
Change the Minecraft rules themselves and make it something all-new!
Think: multiplayer games like **Top Miner**, **Don't Fall In the Lava** and
Banana Race.



LAUNCH THE MODS EDITOR

Login to your Tynker account at **tynker.com** or start the app.

From your Dashboard, click on the **+Create Project** button, then find **Mods** in the **Minecraft** category.

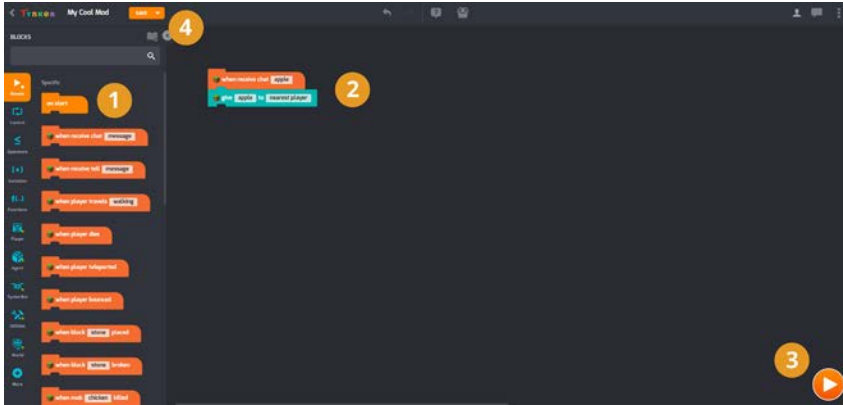


- 1 First: Choose your **Version** of Minecraft
- 2 Scroll through available Modding starter projects in the horizontal scrollbar
- 3 See a **Preview** the project here
- 4 Click **Start Project** to begin editing

For now, just open a **Blank Mod Template**.

THE MOD EDITOR

When you click **Start Project**, you should see the Tynker Workshop appear. But instead of seeing the familiar Stage with actors, you'll see a screen like this:



There are a few main parts of the Minecraft Mod editor:

- 1 This is the **Code Palette**. Blocks are grouped together according to what they do. (We'll explore each category in turn, next.) To find the block you need fast, you can Search using the .
- 2 This is the **Coding Area**, where you drag and drop code blocks from the Palette. Connect a stack of code blocks together to write your own programs. You can have as many scripts as you need, but only code blocks that are connected will run.
- 3 Ready to try it out in Minecraft? Use **Play**  to deploy your code. Follow the on-screen instructions to connect to Minecraft.
- 4 **Save your work**. You can change the name of your project from **"Untitled"** to a name of your choice in the upper left hand of the workshop. Then click **Save**. Now, whenever you login to Tynker, you'll be able to see and update your project in **My Projects**! You can also **Save As** or **Publish** your project from this menu.



BTW

If you've done some Tynker block coding, you might be wondering... "Hey! Where is the Stage and Actor List?" While modding, you can think of your Minecraft World as the Stage where you can see the results of your program. Likewise, you must write programs that control Minecraft entities using available code blocks (instead of using an Actor's List to organize your program's logic).



COOL NEW CODE BLOCKS

When coding Minecraft Mods, you have a new set of blocks available in the Code palette. You'll see:

- Familiar programming concept blocks—think about **on start** and other classic code blocks like **math operators** and **wait** or **if-else blocks** that are useful for controlling program flow
- Some all-new Minecraft categories, like **Player**, **Agent**, and **World**—these give you direct control of Minecraft itself!

Let's take a tour of the new code block families.

Events

You can think of these blocks as triggers to run your mod—nearly all of the Minecraft event blocks begin with "**when**." You can run code in response to a certain message in chat, or when a certain action occurs (when an enemy is defeated, for example).

Notice how there are two types of Minecraft event blocks:

- **Specific Events** allow you to choose a particular item or block:



- Then there are **General Events**, which use purple placeholder blocks. These events trigger on ANY block, item, or action:



Control



Control structures like **if**, **if-else**, and **wait** are found in this category. Use them to control which branches of your code execute, as you would in any other Tynker block-code program. If blocks are expecting boolean expressions (conditions that are either true-or-false).

Operators



Operators allow you to create mathematical and logical expressions in your code. Use these blocks in concert with variables to take total control of Minecraft.

Variables



Variables are places to store useful information in your program: think about adding a scorekeeping function or adding the ability to keep track of how many objects a player has collected. We'll explore variables in depth in Chapter 7.

Functions



Functions are user-defined procedures that allow you to organize your code. Think of functions as “writing your own custom code block.” We’ll explore variables in depth in Chapter 7.

Player



The Player blocks allow you to apply effects, teleport players, and give players items.

Agent



A robot within the Minecraft world that you can control. Minecraft agents can mine, build, farm, and more. They even have their own Inventory! Think of your agent as a Minecraft mob that you can control with code.

TynkerBot



TynkerBot is a special “superfast” Minecraft agent. Think of it as an invisible cursor in the Minecraft world, with superpowers.

Utilities



These Tynkerbot-powered Utilities give you primitive building shapes like boxes, cylinders, and spheres. You can also summon mobs or teleport players with Utilities Blocks.

World



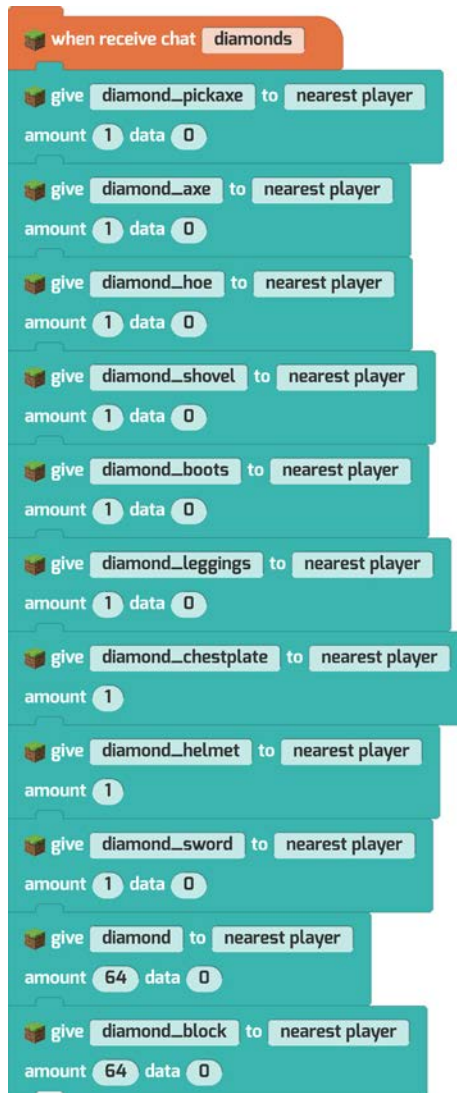
The World code blocks allow you to change aspects of the game itself, such as the Game Mode (from Creative to Survival, and vice versa), changing the weather, and abilities to set blocks at a certain location.



PROJECT #1: GIVE YOURSELF ANYTHING!

Let's complete a short program to get the big idea. What is something you've always wanted to obtain in Minecraft? Think for a minute.

Maybe it's a full Diamond Armor. Here's the code that lets you get everything Diamond-related at once:



Try writing your own program to give yourself items and blocks. The sky's the limit! You're only limited by what you can carry....

To deploy your code, press the **Play** button. A connection code will appear: copy and paste the connect code into chat. Then enter **"diamonds"** to trigger the code in chat.



Bonus Task

Write a second script that allows you to enchant your sword with amazing effects. Trigger it with a different chat command. Here's one idea:



PROJECT #2: IGLOO DWELLER

An igloo is a spherical dwelling, invented by the Inuit to keep warm during arctic winters. Despite being made out of snow, the dwelling is sturdy and provides excellent shelter from harsh weather.

You can build your own igloo using some TynkerBot magic:



Try programming it yourself, then trigger the code by entering “igloo” in Chat. This hemisphere is a great start at a cool domed house. Add a door, some torches, and you’re nice and cozy.

And notice how you can control the parameters of this function, too. Customize how they work, using dropdown menus!

To deploy your code, press the **Play** button. A connection code will appear: copy and paste the connect code into chat. Then enter “**igloo**” to trigger the code in chat.

Try practicing with other shapes in the **Utilities** category to get the big idea: You can make different shapes that are **filled** or **hollow**, and you choose what kind of block to use, too.



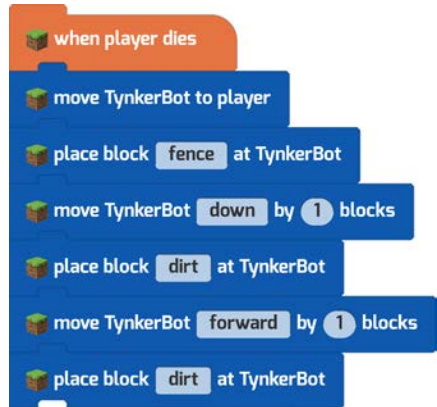
Bonus Task

What kind of shape is your perfect Minecraft house? Try some other shapes, like **Box** and **Cylinder**. Try using loops and other commands. You might need to use pen and paper or another planning tool to see how all the different “parts” of your house fit together perfectly.



PROJECT #3: GRAVE MARKER

If you play Minecraft long enough, you're bound to **die** at some point or another. Thankfully, you always respawn. This simple script reacts to a player's death by marking the spot where they fell.



Notice how the TynkerBot moves to the player, then builds a sequence of blocks that mimic a grave marker.





Bonus Task

Experiment with TynkerBot commands to get the hang of it. Sketch out a design and make your own monument! Try exploring the other Minecraft events that can trigger a custom build. BTW, don't forget to collect your gear after you try this one out.



PROJECT #4: FLAMING BOW

Want to enchant an item? It's simple. Here's a script to give the player flaming arrows!

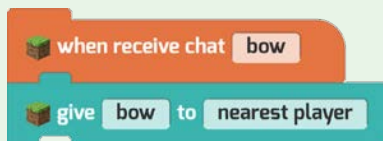


Check out the results in your Minecraft game:



✓ **Bonus Task**

Don't have your special item already in your inventory? Try a program like this one:

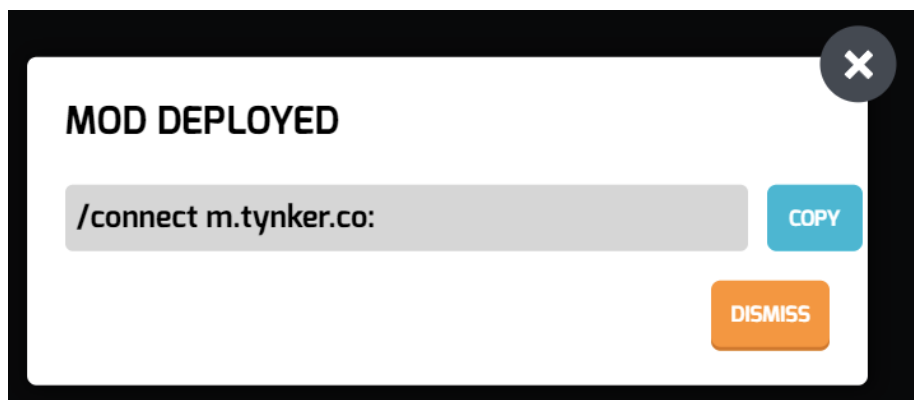


DEPLOY YOUR MOD

OK, you've done it! You've written your first mod in Tynker and you want to try it out.

Here's the big idea:

1. Write your mod in Tynker, then press **Play**—a popup will appear. You can think of this command as your “**Connect Code**.” The **copy** button allows you to copy this address to your clipboard—you can also write it down carefully. This is how you'll connect to Tynker from your Minecraft World.



2. Open Minecraft. Then **Create** a new Minecraft world or open an existing one. Make sure that **Cheats** are enabled while creating it. **You must allow cheats in order to connect to Tynker's servers.**

Hint

By default, games in Creative Mode have cheats enabled.

3. When you're within your Minecraft world, open up Chat (press T). Then type in your "connect code" carefully—or paste the "connect code" if you copied it to your clipboard. Double-check your spelling: Forget a single character, and it won't run!
4. Press **Enter**, and the command will execute. If you're successful, you'll see a connection message appear in chat. You should see your mod in the world right away. If your mod is activated by a particular Chat message or event (obtaining a resource, breaking a block, and so on) make sure to perform the action in question.

A screenshot of the Minecraft chat interface showing a message that says "Connection established to server! ws://m.tynker.co/m-". The text is in a pixelated font, typical of Minecraft chat.

Once you're within your Minecraft world and connected to the Tynker servers, you can tweak your code and re-deploy on the fly. Just make some changes to your code, press **Play** in Tynker, then switch back to your Minecraft world. You should see a message in Chat assuring you that your mod was recently updated.

Having difficulty getting your first mod to deploy? Flip ahead to the Troubleshooting Appendix. ***The most common error is misspelling the URL.***

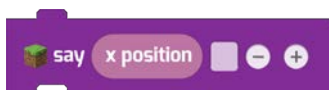


ONE FINAL NOTE: ON DEBUGGING

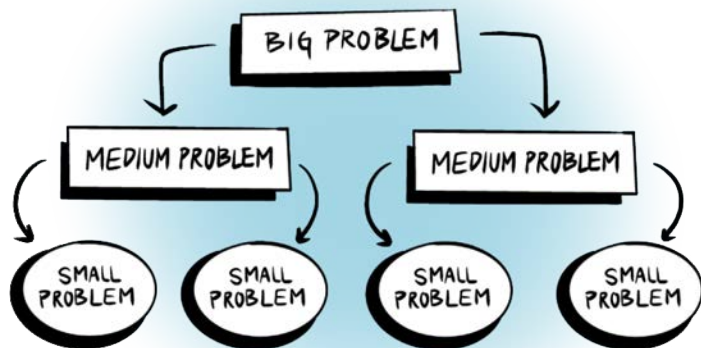
As you start to write your own original mods, you'll find that sometimes your programs won't work exactly as you expected—and sometimes they might not work at all! The process of finding and fixing bugs in code is called **debugging**. Ask any computer programmer, and you'll find that debugging is a big part of their day-to-day work.

If you've been coding for a while, you've probably done your share of debugging, too. Here are some tips from the Minecraft team at Tynker on finding and fixing bugs in your code:

- Report values using chat—for example, here's how you could say the value of the variable called **x position**



- Break big problems into smaller ones—test your code as often as practical

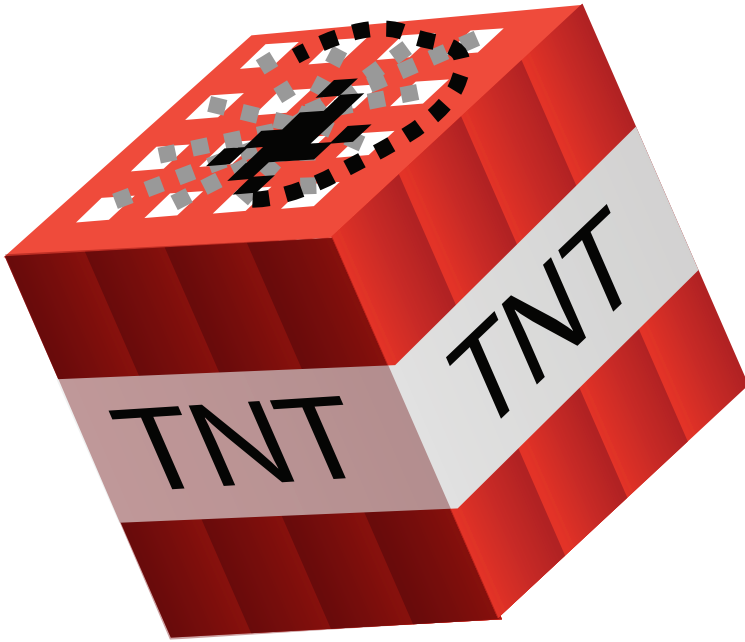


- Write code a little bit at a time—run your code as you add features, rather than writing everything all at once.
- Plan out and sketch—write “pseudocode” describing the steps you want your program to achieve.
- Save different versions of your code before making big changes, in case you need to revert. Use **Save As** to save a different version of your Tynker project.
- Experiment in a new project, testing a small piece of your code—use Tynker’s **Backpack**  to save code so you don’t have to rewrite your script.
- Ask a teacher or another coder for help.
- Get a good night’s sleep and try again tomorrow! The best debugger is a well-rested mind.



BTW

Remember that programming is hard. If at first you don’t succeed, try, try again—and think “smaller” if you have to. Be patient with yourself and work systematically.



Chapter 6

Modding Projects

Explore key Minecraft concepts like absolute and relative positions, as well as block data values. Learn how to teleport the player, summon mobs in a flash, and more.

Start at tynker.com/modbedrock



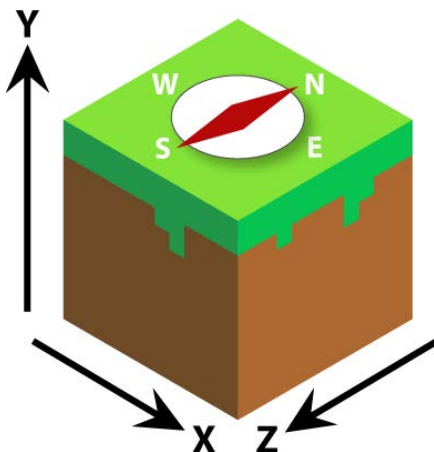
LOCATIONS IN MINECRAFT

There are two different ways of representing locations in the Minecraft world.

Absolute Position

Because the Minecraft world is 3D, you must use 3 axes, X, Y, and Z to represent a single position.

- The X-axis runs West-to-East
- The Y-axis describes vertical height (sometimes called elevation)
- The Z-axis runs North-to-South



Representing a single location in this way is called its **Absolute Position** or sometimes its **World Position**. For example, here's one code block that uses an absolute location.



This code block teleports the TynkerBot anywhere you like. Just update the numbers in the block.

Hint

You can display your current coordinate position within the Minecraft world by entering this command into chat (open chat by pressing T) **/gamerule showcoordinates true**. You should see something like this appear in the upper left:

Position: 139, 63, -4

Relative Position

But more often than not, you'll also want to use coordinates that are relative to a player's current position. For example, if you wanted to build a fence that was 5 blocks south of the player's current position. Or dig a hole, straight down! (Since players are two blocks tall, these locations are relative to the player's feet.)

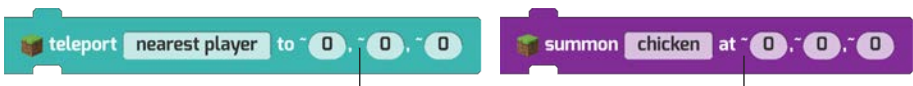
Relative location is indicated in code using the squiggly line ~, which is called a tilde (till-deh). The notation for coordinates looks like this:

- ~X ~Y ~Z

You can think of relative location as an "offset" or a "change in" location in a certain direction, where:

- A positive change in **X** is moving East (negative is West)
- A positive change in **Y** is moving Upwards in elevation (negative is Downward)
- A positive **Z** is South (negative is North)

You'll see relative coordinates in code blocks like these:



The squiggly line indicates that these are relative locations

Hint

Notice how these blocks and all relative locations are relative to the cardinal directions (that is, North, South, East, and West, rather than the direction your player is currently facing).

MINECRAFT DATA VALUES

Let's cover one more important Minecraft idea, before writing some mods:

Minecraft Data Values

Many Minecraft blocks have an additional setting called a data value. These **data values** specify variants of a single block. In most cases, these variants are simply different colorings. For example, here are some data values of Wool.

-Name-	-Data Value-	-Name-	-Data Value-
 White Wool	0	 Light Gray Wool	8
 Orange Wool	1	 Cyan Wool	9
 Magenta Wool	2	 Purple Wool	10
 Light Blue Wool	3	 Blue Wool	11
 Yellow Wool	4	 Brown Wool	12
 Lime Wool	5	 Green Wool	13
 Pink Wool	6	 Red Wool	14
 Gray Wool	7	 Black Wool	15

Notice how wool with a data value of 0 is White. However, wool in Minecraft can also be Orange (data value 1), Pink (data value 6), Purple (data value 10), and so on.

By default, blocks have a data value of 0—that is their most common variant.

You can specify a data value in the field marked **data** in Tynker code blocks.

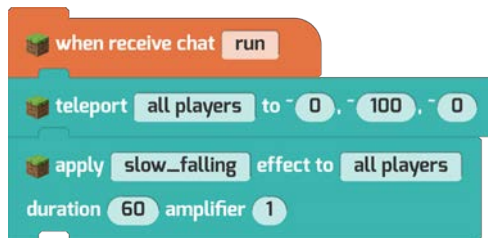


This code block will give the player 5 blocks of Black Wool.

PROJECT #5: FREE FALL

Let's try a simple script that makes use of relative locations. Recall that the Y axis in Minecraft controls an object's elevation.

Here's a script that teleports all the players 100 blocks higher than their current position, then gives them a Slow Falling effect. Doing this ensures all players can enjoy the "bird's eye" view of the world as they float down—and no one takes any fall damage too!



Try changing the location specified in the blocks to teleport players into a completely different location.



PROJECT #6: ENDLESS BLOCKS

Here's a short script that makes use of the special "**General**" event blocks, as well as what you learned about data values. Read the code and then try it out:



Notice how instead of being triggered by the **when gold block placed** command (or another particular block-type), instead, there is a purple placeholder block. This code gives the player back whatever block they just placed!

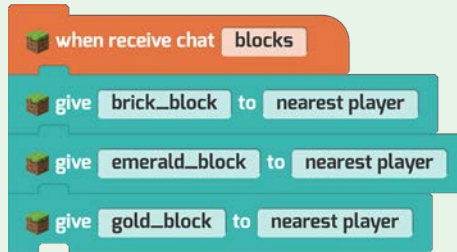
As long as you have one block of any type, you now have an infinite supply!

The if-else block handles specialized block data values: If there's a data value associated with the block that was just placed, you get the exact same block back (that is, a block with the same data value—for example, placing Black Wool gives the player Black Wool). Otherwise, you get a block with a data value of 0, which is the default block.



Bonus Task

Starting from empty? Give yourself a few starter blocks if you need to, like so:

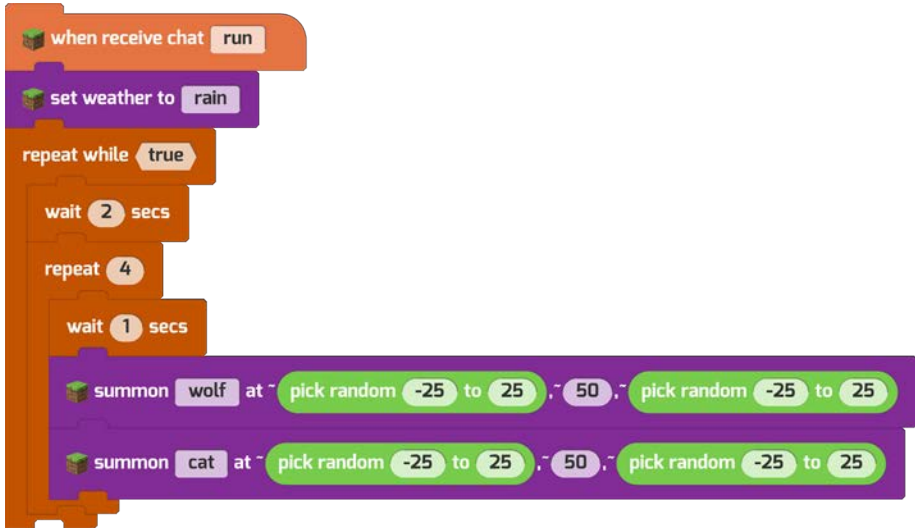


PROJECT #7: RAINING CATS AND DOGS

Let's take what you learned about relative location and apply it to a mob-summoning program. Imagine you wanted it to literally rain cats and dogs—that is, have the mobs fall from the sky when it rains.



Here's how you can do it.



First, we set the weather to rain, then within a loop with some **wait** blocks, we summon animals using relative location. The wait blocks ensure that the animals aren't summoned too fast, which can slow down your computer!

Notice how the code uses randomness and relative locations to put the animals in the right place. They're always 50 blocks above the current position, but they're placed randomly in the X and Z axes, between two possible values.

Take a closer look at the **pick random** block used in the program. With each run of the program, you'll get a new value between the range given.

pick random -25 to 25

How else could you use the summon or spawn blocks to make interesting things happen? Maybe you want it to rain cake instead. Try something of your own imagining.



PROJECT #8: SUMMON A BOSS

Here's a fun idea. Use what you learned to invent a custom boss using Tynker's **Mob** Editor, then write a program in Tynker **Mod** Editor to summon it!

Flip back to Chapter 4 for a refresher on how the Mob Editor works. We chose to reskin a Snow Golem for our boss, giving him extra size and health. Here's what it looks like in Minecraft. Pretty scary!

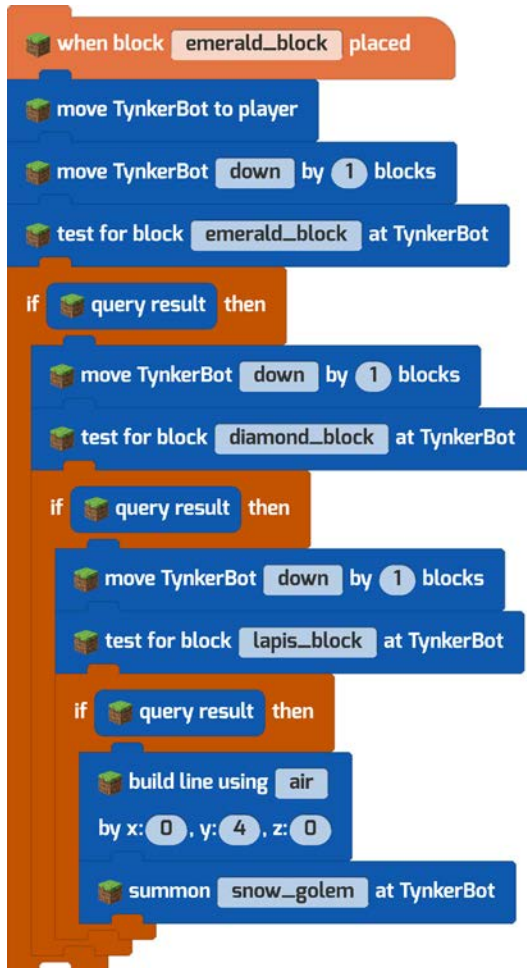


Once your mob is complete, deploy your new **Resource Pack** to your world. Now, if you want to play against your new boss, you could use the boring-old **summon** command in Chat.

But let's use Tynker to create a custom mod so that you can summon your boss by placing a block formation in game, just like Minecraft's Wither! Switch to Tynker and open the Mod editor (**Create Project > Minecraft > Mod Editor, Bedrock**). Now it's time to write a custom script to summon a Snow Golem.

In this example, the player must build a totem of 3 blocks to correctly summon the boss:

- Emerald
- Diamond
- Lapis



Notice how the code uses **query result** and TynkerBot to test for the correct sequence. If everything is in the right order, a snow_golem is summoned! Get ready for a big fight!

Two important things to note: You must be standing on top of the totem as you build it in order to summon the boss, since the TynkerBot is summoned to the player and starts moving down from that location.

Additionally, notice how the blocks are checked (and programmed) in reverse order of how you must build it—with the **when block placed** event using the final block of the totem.



Bonus Task

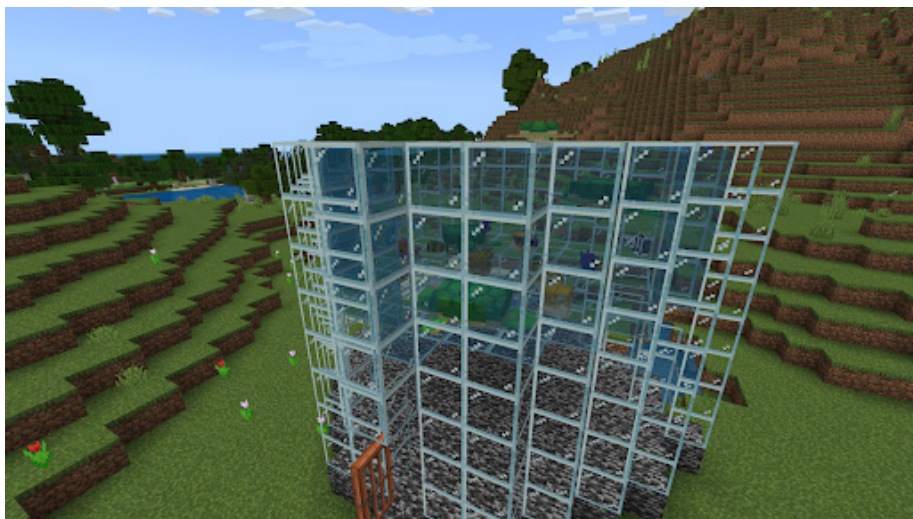
Try changing the totem “spell” to summon your boss. Try other ways of making villains or bad guys.

Make sure to give the player a good reward for defeating the boss, too.



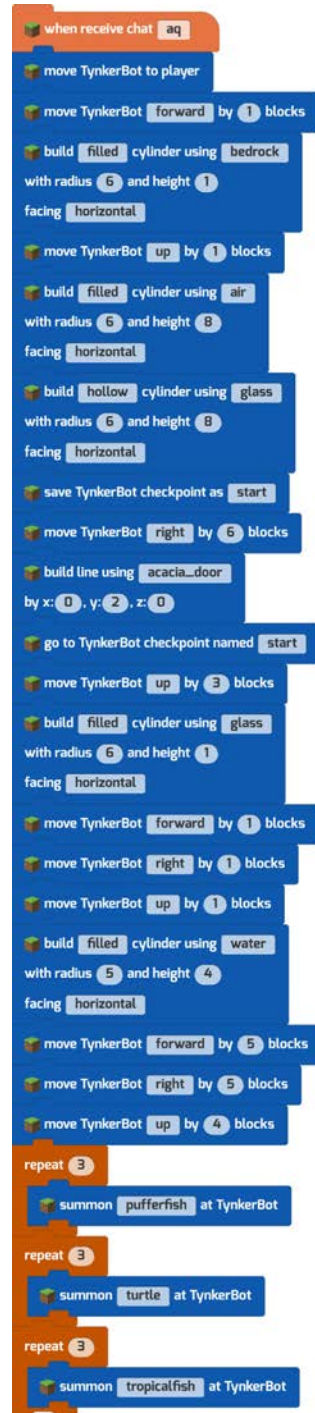
PROJECT #9: AQUARIUM

In this project, you'll use some TynkerBot magic to build a larger-than-life aquarium! The script is long, but the idea is simple.



Here are the steps for building the aquarium:

- Move the drone in front of the player
- Build a 1 block tall layer of bedrock for the floor
- Move the drone above the floor and fill the area with air (to get rid of anything in the way)
- Build the walls of the aquarium, a hollow cylinder of glass
- Move the TynkerBot to the right by the **radius** of the aquarium and build the door
- Move the TynkerBot up and inside of the aquarium and build the bottom of the fishtank out of glass
- Move the TynkerBot above the bottom of the tank and fill it with water
- Lastly, move the TynkerBot inside the water so we can spawn fish at its location



Read the code carefully to get the big idea. As a finishing touch, after filling our cylinder with water, we summon some nice fish for the tank. Three pufferfish, turtles, and tropical fish are summoned at the TynkerBot, and they'll then swim around their lovely new aquarium.



Bonus Task

Change up the aquarium so it's a new shape! What about coding a petting zoo?



Chapter 7

Minigame Mods

See practical applications of computer science concepts like variables, lists, and functions, as you create your own original minigames. Learn how to keep score, organize your code, and much more.

Start at tynker.com/modbedrock

We'll explore the idea of **abstraction** in programming. Abstraction and some other coding tricks can help you improve the readability—and the re-useability—of your Minecraft mods.



Ready for a fun interactive Minecraft course, designed to teach beginners from the ground up? Skip ahead to **Agent Recall**, and try to escape from the enchanted castle, using your code blocks to solve puzzles.



ABSTRACTION IN EVERYDAY LIFE

Imagine you're texting with your friend.

what are you up to? 😊

just finishing homework,
then will crash

Your friend Jen just said she plans to “crash” tonight. The word crash is used here as an abstraction. Abstractions let us focus on the big picture. And just as importantly, abstractions let us ignore the details.

Your friend saying they are going to crash gives you all the meaning you need for this context. It is efficient and to the point. It would not add anything for your friend to say: “I plan to take a shower, change into pajamas, brush my teeth, lay in bed, then read a comic book...” and so on.

In computer science, it is common for us to use abstractions to refer to our algorithms and data simply and efficiently. We use abstractions to give simple names to:

- **Variables**—Keep track of data over time in your programs
- **Functions**—Make your own “custom blocks”—these custom procedures help keep your program organized, and let you avoid writing the same code over and over.

VARIABLES IN EVERYDAY LIFE

Imagine you want to make a bread-baking simulator Minecraft mod, inspired by cooking games like *Overcooked* or *Burger Time*. You want to be able to count the player’s total amount of bread produced. But.... you can’t just use the player’s inventory (since bread can be eaten or thrown away). Let’s use variables instead.

Variables look like this in Tynker.

bread

x position

Try creating your own in the Coding Palette to get some practice. Click on **Variables**, then choose **CREATE VARIABLE**.

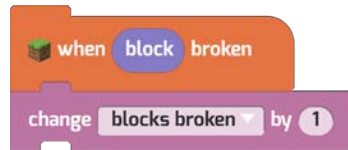
Code can change the value of a variable, so that you can keep track of a value over time. For example, here's a simple bread-tracking script:



You can also **set** the value of a variable. Use this block to update the value, too. You can reassign the value of a variable at any time!



Here's a simple block counting script, using a custom variable named **blocks broken**. Notice how this script will count when ANY block is broken in-game.



Can you figure out how to use the **say message** block to report when the player hits particular milestones, like 100 or 1000 blocks broken?

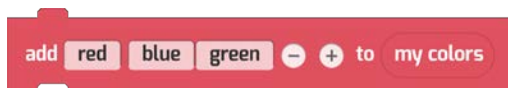
Hint

By default, Minecraft variables are **global**, which means that any script you write can access ANY variable. Script variables are also available, which are local to a particular stack of code blocks.

LIST VARIABLES

List variables are another useful way of organizing your program's data. Instead of keeping track of a single value, you can use **list variables** to keep track of a list of items, in a particular sequence.

You can think of this variable type as a little bit like a "shopping list". Here's how you might add 3 elements to a list called **my colors**.



Elements inside the list can be accessed (or changed) by **index**, a number that refers to the item's location:

ELEMENT OF MY LIST	RED	BLUE	GREEN
Index	1	2	3

You can add new elements to a list like so, and **purple** would take the index value of **4**:



This special block returns a random element from the my colors list. Pick a different color at random!



You can create Lists yourself using the **CREATE LIST** button in the **Variables** code palette. You'll get some practice with lists in the final project of this chapter, and by taking Tynker's cool Minecraft courses.

FUNCTIONS

Functions let you organize your own code. For example, imagine a game that requires a reset, which moves objects or actors to a certain location, again and again. Rather than copy-paste your code each time you need to reset an object, you can write a function.

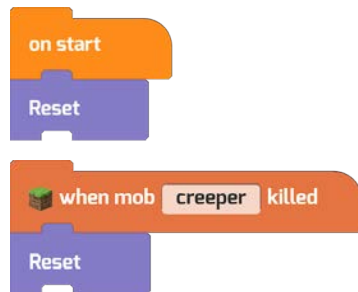
First, create a new function by going to **Functions > Create Function**.

Name it **“Reset”**. Then define what you want your function to do, using code blocks:



By attaching blocks, you have now **defined** what your function *does*. Then you can **call** your new function as many times as you like, and the code will run.

Here are some sample programs, calling the reset function.



Functions can take parameters too. For example, if you write a program to spawn sheep at a particular location, you might add a parameter so you have more flexibility in the overall number of sheep are spawned. You can add parameters as you Create a function. But you can click on the pen  in the function definition to add new parameters to an existing function, too.



FUNCTIONS CAN CALL OTHER FUNCTIONS



BTW

You can call functions within other functions! Here is a function that calls four other functions to draw a skyscraper.



COMMENT YOUR CODE

One final tip to help keep your programs organized: A **comment** is a human-readable description of the code in a program.

Good programmers add comments to make their code easier to understand by other programmers or even users of your programs. Tynker's comments look like this:

```
// enter "house" in chat to build a HUGE house
```

Are you making a big project but don't have time to finish coding it in one sitting? Add comments to your code, reminding you of what needs to be done or isn't working yet.

```
// finish this scorekeeping code later
```

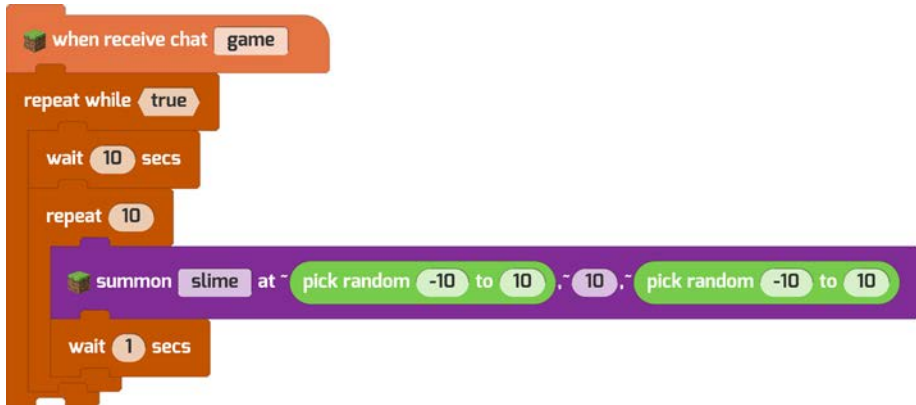
PROJECT #10: SLIME SMASHER

Let's try to make a fun minigame that uses variables to keep score. We'll summon loads of slimes near the player, and see how long they can last, keeping track of how many slimes they defeat. Here's what this hectic, simple game might look like in action:



To follow along in Tynker, create a variable in the **Variables** palette, then choose **CREATE VARIABLE**. Name it something like “**slimes**” or “**slimes-killed**.” You should then see the block available to you.

Here's the code to summon an endless stream of slimes, all around the player, falling from 10 blocks high. Notice how it's triggered by the message "game" in chat, and how the range of X and Z is -10 to 10 blocks away from the player. Slimes are always summoned 10 blocks above the player (a constant offset of positive 10 in the Y direction).

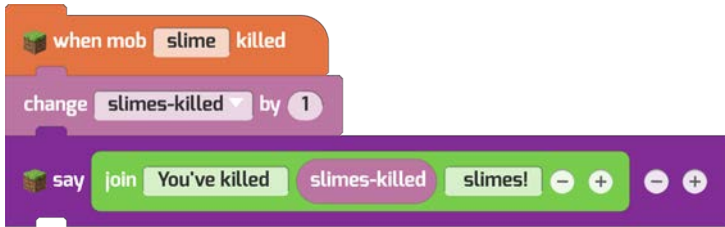


This script sets the initial value of **slimes-killed** to 0, which is how we want our game to begin.



The final script for this game adds 1 to the **slimes-killed** variable, whenever a slime is killed. The player also receives a message in chat, letting them know exactly how they're doing! Notice how the **join** block is used together with the variable to report its value in a user-friendly way.

```
[External] You've killed 18 slimes!  
[External] You've killed 19 slimes!  
[External] You've killed 20 slimes!  
[External] You've killed 21 slimes!  
[External] You've killed 22 slimes!  
[External] You've killed 23 slimes!  
[External] You've killed 24 slimes!
```



Bonus Task

Try changing the rules of this game to suit yourself. Perhaps you want to give the player a weapon, or summon different kinds of mobs to battle. Can you make the player's score reset to 0 when they are defeated? Or only report the score at certain points, like reaching multiples of 10?

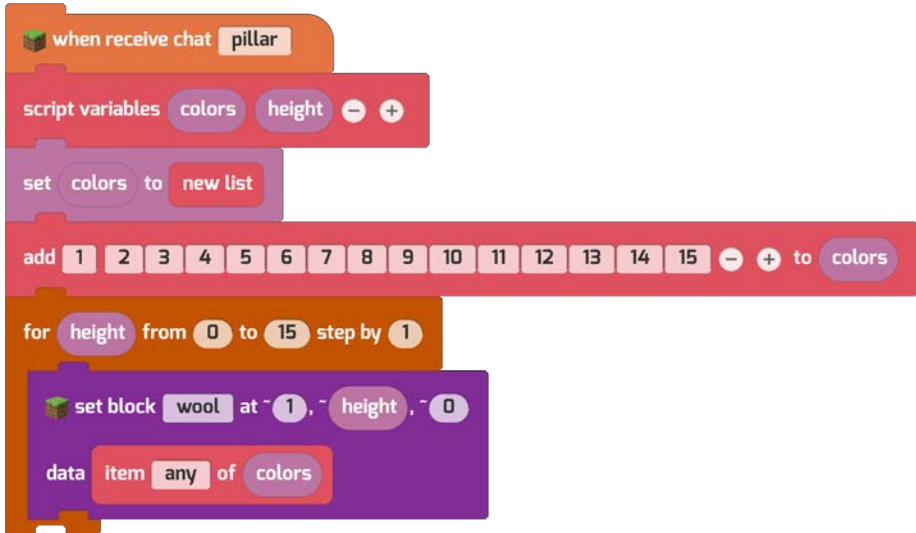


PROJECT #11: RAINBOW PILLARS

Now let's use what you learned about lists to take control of block data values. Imagine you wanted to build colorful pillars in your world a little bit like this:



Here's the code to achieve the effect.



Notice how two variables are created with the **script variables** block, **colors** and **height**.

We define **colors** to be a **new list**. Then we add elements to the **colors** list, from 1 to 15, using the **add elements** block. Finally, using a special kind of loop called a **for-each** loop, we increment through **height**, from 0 to 15, stepping by 1.

The script places a wool block with a 1-block offset in the X direction, but controls the Y elevation using the **height** variable. With each increment of the loop, height's value increases by 1, stacking up our blocks neatly in a pillar. Each time the loop runs, the **item any of colors** block returns a random value from 1-15, making it a different pillar each time it runs. When you try it yourself make sure to try placing a few different pillars using "pillar" to trigger the code in chat.



Bonus Task

Change the list values of colors to be your favorite Wool colors!

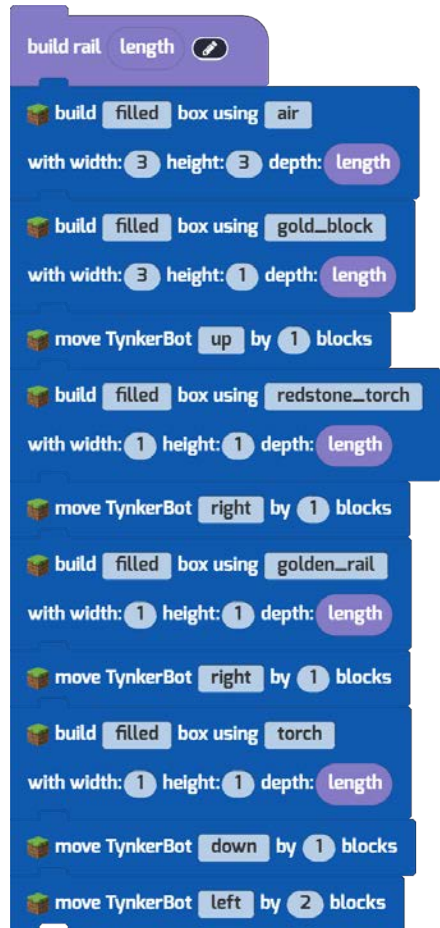
What about creating other, more deliberate kinds of patterns, like a checkerboard or sequence of colors? Lists and loops are perfect for pattern-making of all kinds.



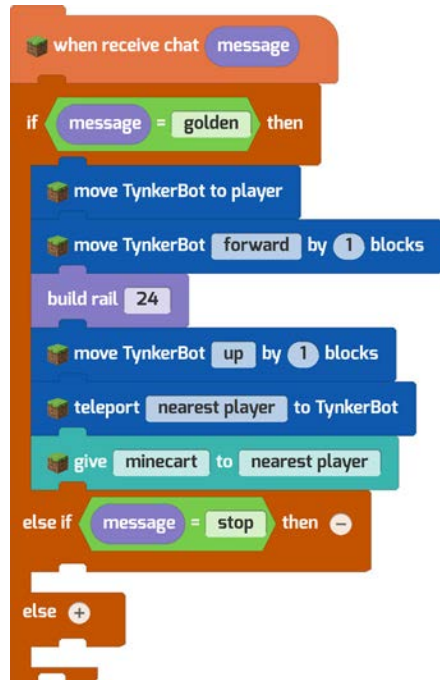
PROJECT #12: GOLDEN RAILWAY

Consider the following code.

Notice how it has a sequence of commands organized into a **build rail** function using a **length** parameter.



Then the player can trigger the function call using chat, using a script like the one below. Notice how when the function is called, the programmer passed the value of 24. In each of the **length** blocks in the function definition above, the value of 24 will be used!



The player is given a minecart, of course, so they can take a ride on their new railway too. The results should look something like this, a golden railway that's lit up and beautiful! Travel in style.



PROJECT #13: HOLE TO THE VOID

The void is Minecraft slang for the empty space that's underneath the surface world.

Let's write a program that burrows deep into your Minecraft world and reaches the void. It's a journey to the center of the earth!



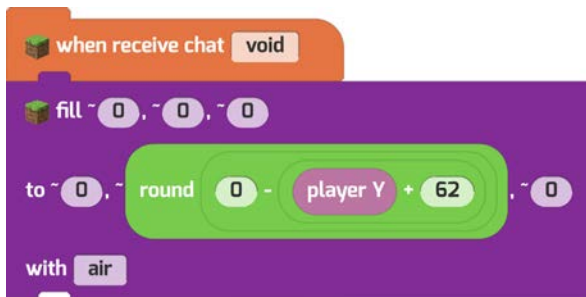
Warning

Players traveling in the void will take damage. Enable Creative Mode for a safe journey (enter **/gamemode creative** in chat). Tunneling in your Minecraft world may release scary monsters.

Here's the first part of the project code:



The player must travel at least one block (so that the **player Y** variable has a value) before triggering this message in chat, "void", which fills an area with air (replacing any blocks along the way).



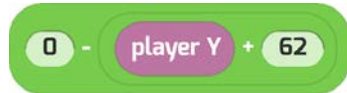
MATH TRICKS

Let's take a closer look at the expressions we used to tell the code how far to dig. Here's a small part of it:


$$\text{player Y} + 62$$

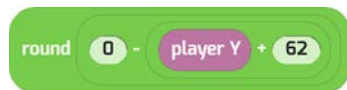
This expression calculates the number of blocks that we need to replace in order to reach the void. Recall that **player Y** will describe the player's current elevation. And since the void is always at the **y-position -64**, this will tell us how far away the player is from the void. (We only add 62 instead of 64 because the player is 2 blocks tall.)

But you might notice that this is a positive number, meaning that it would replace blocks **above** the player rather than **below**. To solve this problem, we use the this trick:


$$0 - (\text{player Y} + 62)$$

This sneaky calculation results in a negative value—allowing us to replace blocks **beneath** us. But there's one more issue that we need to look at.

The player doesn't move 1 block at a time—they move in-between blocks. In other words, you might be standing on the edge of a block, or in the middle, or halfway in-between. This special **round** block makes sure that we have a whole number:


$$\text{round}(0 - (\text{player Y} + 62))$$

This rounds the calculation to the nearest whole number (integer value). So **100.53 blocks** turns into a much more manageable **101 blocks**. That's it! Now you know some seriously cool tricks for working with custom locations in Minecraft!



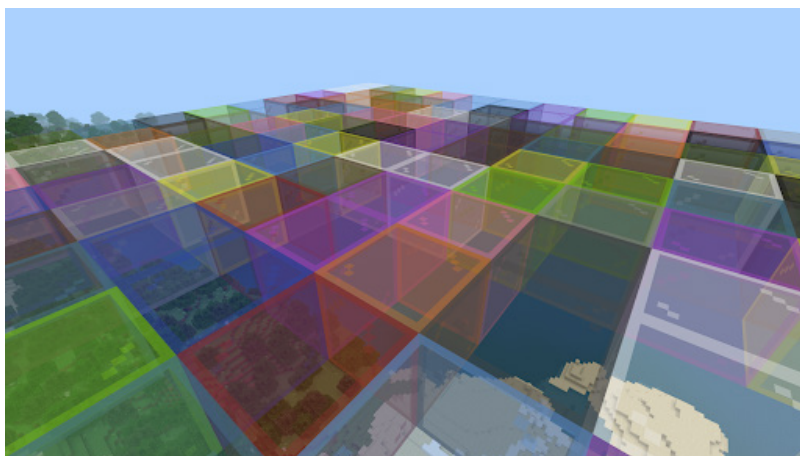
Bonus Task

Try transporting yourself to the void, too! Where else do you want to travel? What other location, building, or teleportation tricks do you want to try?

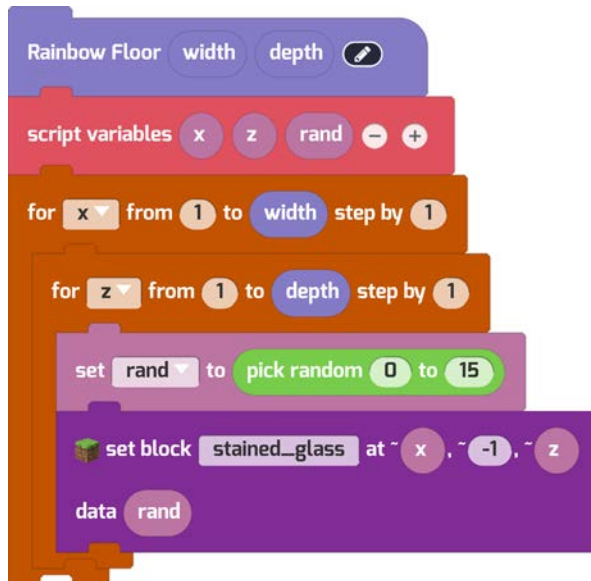
PROJECT #14: RAINBOW DANCE FLOOR

You've probably already noticed how you must carefully embed commands within one another. But you can even nest two **repeat** loops inside one another, to draw a grid. This technique is sometimes called a **nested loop**.

Here's one cool application: drawing dance floors or walls in your Minecraft world!



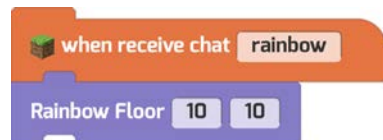
Let's write a function to build a dance floor of any dimension we like. To create a function like this yourself, go to **Functions > CREATE FUNCTION**.



Notice how this function takes two arguments: **width** and **depth**

This function will be flexible. The player will be able to specify just how big their dance floor will be. The function uses script variables, **x**, **z**, and **rand**—the script uses a special for-each loop to move block by block, to create columns and rows. As each block is placed at **x** and **y** locations using the **set block** command, a new value for **rand** (short for random) is chosen. This is used as the data value for the Stained Glass block.

Of course, by itself, this function doesn't do anything yet! We have only **defined** what our function does. You have to **call** your new function using a script like the one shown here. This one will create a dance floor that is 10 × 10 blocks!





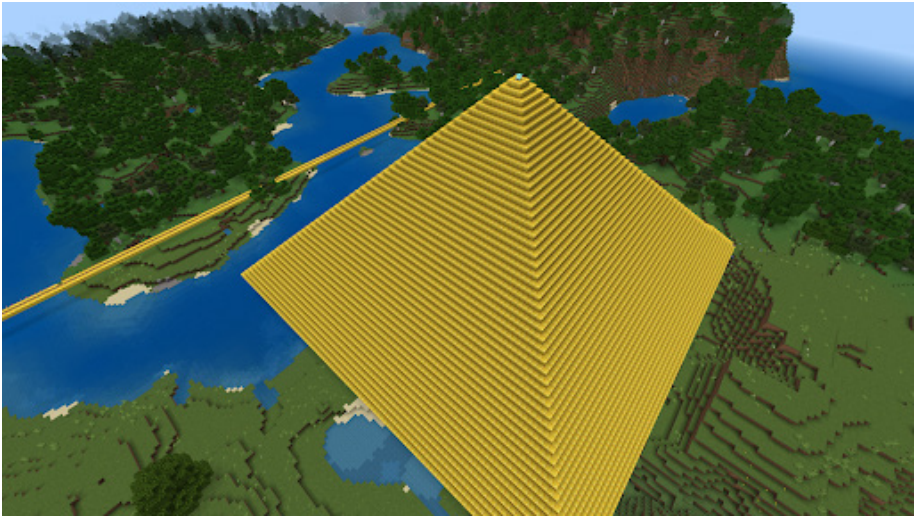
Bonus Task

Try creating a checkerboard using the mod operator. What other tasks could you solve with nested loops? Try building a farm or walls around your house.



PROJECT #15: PYRAMID

Let's try another application of nested loops to build a real Minecraft classic: The Pyramid!

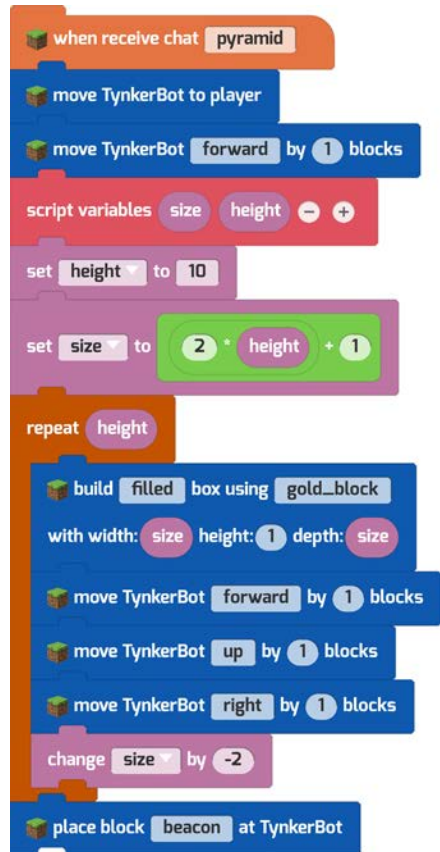


We used gold blocks for ours, but try changing the code so it's built using your favorite block. Maybe it's time to make a melon block ziggurat!

Here's the code that does it.

Notice how the commands use two script variables, **size** and **height**.

Height determines how high the pyramid will be, while the **size** specifies both **width** and **depth** (this means that each layer is a square). The code builds each layer of the pyramid from the bottom up. The **size** of each layer decreases by 2 with each repetition.



Bonus Task

Try re-organizing your code into a build pyramid function to make it even more flexible and powerful (**Functions > CREATE FUNCTION**). What else could you create? How could you make a **hollow** pyramid?



Appendix A

Where To Go From Here?

Looking for more Minecraft resources? Interactive courses? Find a complete course listing, and see what's compatible with Bedrock AND Java! Find it all here.

Start at tynker.com/modbedrock

Looking for more starter projects on Tynker? Login at **tynker.com** or open the Tynker app, then choose +Create Project and look inside the **Minecraft** category. Find a project that sounds interesting, and try it out!

TYNKER'S MINECRAFT COURSES

Ready to take the plunge? You might also enjoy taking a dedicated Minecraft coding course. Here's a list of everything that's available.

- **Agent Recall**—Enter an immersive Minecraft world and escape from the enchanted castle. Compatible with Minecraft Bedrock.



- **Training Day**—Enter an amazing Minecraft world in this adventure. Create your own Minecraft skins, items, blocks, and more.



- **Gears of Time**—The immersive adventure continues. Download an interactive world with **advanced coding challenges** and follow along. Compatible with Minecraft Bedrock.



- **Mythicraft**—Build Minecraft mobs with custom behaviors while learning about monsters and ancient gods. Compatible with Minecraft Bedrock. Minecraft Mob Editor.



- **Mind Crafters**—Use logic, loops, variables, and functions to bring complex Minecraft structures to life while learning to build 24 cool mods. Compatible with Minecraft Bedrock. Minecraft Mod Editor.



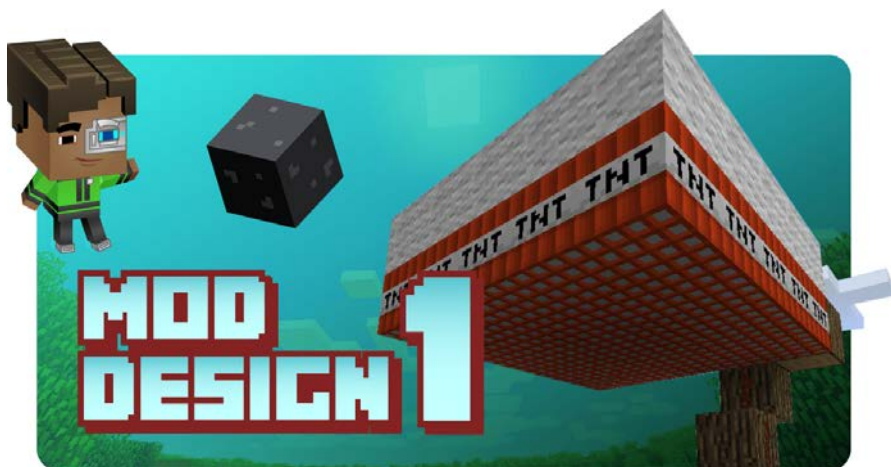
- **Mod Starter Pack**—Make your own skins and experience basic Minecraft modding with Tynker. Access with a FREE Tynker account.



- **Game Design 1**—Learn to create 3 classic games in Minecraft—Capture the Flag, Slimeball Soccer, and Tower Defense. Compatible with Minecraft Classic (Java).

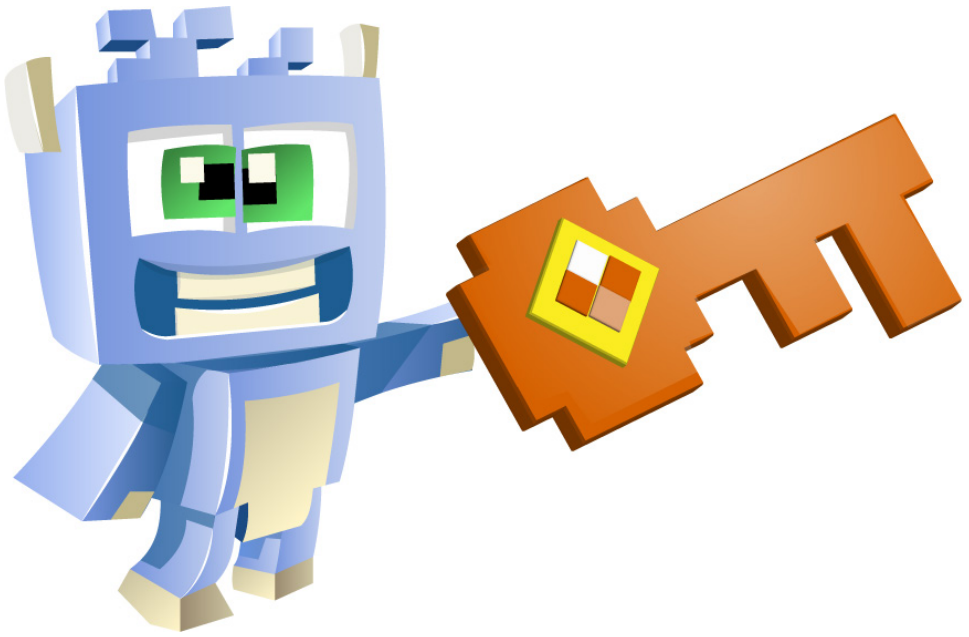


- **Mod Design I**—Learn to mod Minecraft with Tynker! Build 25 cool mods to host on your own private server. Compatible with Minecraft Classic (Java).



- **Mod Design II**—Create exploding snowballs, a treehouse, a fireworks show, and many more mods! Compatible with Minecraft Classic (Java).





Appendix B

Glossary

Totally new to Minecraft? Learn some of the key terms. And look up key computer science concepts, too.

Start at tynker.com/modbedrock

Absolute Position A set of X, Y, Z coordinates that represents a single physical location in a Minecraft World. See also **Relative Position**.

Agent A robot within the Minecraft world that you can control. Minecraft agents can mine, build, farm, and more. They even have their own Inventory! Think of your agent as a Minecraft mob that you can control with code.

Bedrock Edition The latest and greatest version of Minecraft, sometimes nicknamed “Minecraft for Windows 10” or “Minecraft Pocket Edition.” The alternative is Minecraft Java (sometimes called Minecraft Classic). When using Tynker tools for editing skins, items, mods, or mobs, make sure to select your preferred version of Minecraft before beginning.

Blocks The fundamental “unit” of Minecraft. You can mine blocks, craft blocks, and place blocks. By editing a block using Tynker’s Block Editor, you can change **how blocks look**—and change the texture of your world.

Crafting Creating new items via raw materials—crafting can happen in your inventory or at workbenches. For example, a Gold Ingot can be crafted from 9 Gold Nuggets. Look at your **Recipe Book** for ideas when crafting, or search online for “Minecraft crafting recipes”

Creative Mode A different way to play. Give yourself powers and invincibility immediately: Double-jump to fly around your world. You have infinite resources and blocks, too. Get creative! Switch modes by using the **/gamemode creative** or **/gamemode survival** command in chat.

Chat More than just a chatting function, Minecraft Chat can be used as a trigger for your mods, to report the results of programs, and much more. Press **T** to enter chat

Item Usable in-game objects. Any object in a player’s Inventory or in their hands is an item. Some familiar examples of items are: Apple, Diamond Pickaxe, and TNT

Java Edition (Minecraft Classic) The first version of Minecraft that was ever released. Since the Java Edition was released, Minecraft was rewritten entirely as Minecraft Bedrock. Different versions of Minecraft are compatible with different courses on Tynker. When using Tynker tools for editing skins, items, mods, or mobs, make sure to select your version of Minecraft before beginning.

Mob “Mob” is Minecraft slang for animals, enemies, and anything else that moves. Think: Creepers and Endermen, but Pigs and Sea Turtles, too.

Mod (n.), Modding (v.) An extension or “change in the rules” from normal Minecraft. Some popular mods include Spleef, Build Battles, and more. You can use mods that other people have written, or learn how to code your own.

Relative Position A location in the Minecraft world that is described using a distance in particular direction from the user’s current location. Relative location is indicated in code using the squiggly line ~, which is called a tilde (till-deh). See also **Absolute Position**.

Skin A player’s in-game appearance.

Spawn When items appear in the user’s Minecraft world.

Summon When creatures (mobs) appear in the user’s Minecraft world. Want to make a particular animal or enemy appear? Try using the command: **/summon panda**, replacing “panda” with whatever mob you like.

Survival Mode The normal way to play. You can die—you must fight or avoid enemies to live. Build shelter and survive, if you can! Switch modes by using the **/gamemode creative** or **/gamemode survival** command.

Teleport Move the user or Agent from their current location to another location in the Minecraft World, instantaneously.

Tynkerbot A Minecraft agent that can be controlled by Tynker. Tynkerbot has special abilities above and beyond regular agents, though.

KEY COMPUTER SCIENCE TERMS

Algorithm A step-by-step set of instructions for a particular task, given to a computer. You can think of an algorithm as a little bit like a recipe 

Code A language that computers can understand. Code is how you'll tell a device (like a computer, tablet, or iPhone) what to do

Code Area Where you place code stacks to run, in Tynker Workshop.

Code Palette The searchable list of all code blocks available to use.

Command An instruction that tells a device to do something. Tynker's code blocks are commands

Comment Text in the body of a program designed to be read by a human, not a computer. Comments aren't run by the computer. You can leave comments to yourself as a reminder of what you were working on—or as hints to others looking at your code 

Condition A logical expression that evaluates to true or false. For example, a condition might be, "Does the player have more than 9 stone blocks in inventory?" or "Is the player currently sneaking?"

Conditional loop A loop that repeats one or more commands as long as a condition is met. Example conditional loops include **repeat until** and **repeat while**

Conditional statement A way for your computer to "make decisions" on its own. A type of statement that executes different parts of the code based on whether a condition is true or false. Example conditional statements include **if** and **if-else**

Counting loop A loop that repeats one or more commands a number of times

Degree A measurement of an angle. 90 degrees is a right turn. 180 degrees is turning around exactly. 360 degrees is turning in a full circle 

Debugging Professional programmers make so many errors, all the time, that they have a special name for them: Bugs. Debugging is the process of finding and fixing bugs in your code. 

Event A way to start a particular set of commands. Examples of events include **on start**, **when item used**, **when receive chat message**, **when player dies**, and similar blocks 

Expression Any sequence that evaluates to a single value. For example, the expression $(5-4) * (3-2)$ evaluates to the single value 1. You can use complicated expressions in your code to make it more powerful and flexible

Function A sequence of commands that can be run together as if it were a single command. Functions are a helpful way of organizing your code. Create your own by going to **Functions > Create** in the Tynker code palette. You first *define* what the function does, then you can *call* the function to make the code run.

Infinite loop A loop that repeats forever and does not end until the program stops

Iterate To repeat a code instruction, for example, in a **loop**

Lists A list is a special kind of variable that holds a series of items called *elements* in a particular order. You can store text or numbers in a list! 

Loop A loop repeats one or more commands over and over

Operand The numbers, variables, or expressions being operated upon (see **Operator**)

Operator Mathematical and logical functions like division, addition, and so on 

Parameter An extra piece of information that is passed into a **function** to make it more powerful or flexible. To add a parameter to an existing function, double click on its name in the Code Palette.

Random number When all possible number choices have an equal chance of being selected 

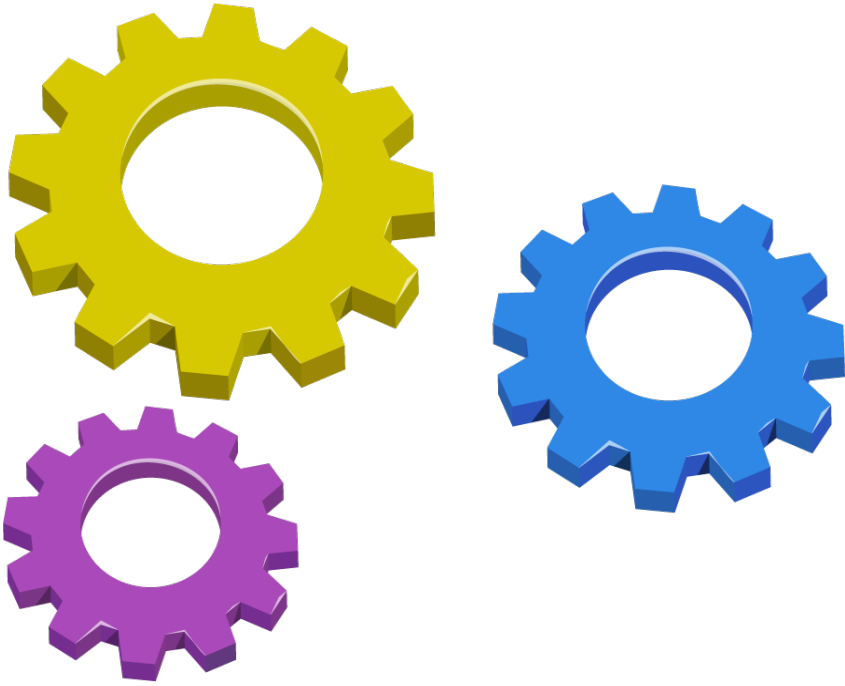
Run To start a Tynker project, hit the Play button  Then follow the on-screen instructions to connect your Minecraft world to Tynker (instructions differ, depending on your version of Minecraft). If any of your programs are triggered by chat messages, you must type those too, after connecting to Tynker (for example, any programs that use the **when receive chat message** block)

Sequence The order in which events happen

Strings Text-based information like “Game Over!” or “Hello World!” or “B”

Variables A value (or piece of information) that is stored by a computer program. To create a variable, go to **Variables > Create** in Tynker’s Code Palette.





Appendix C

Troubleshooting Tips

Nobody said coding was easy. Learn some tips for troubleshooting common errors, learn how to ask for help, and more.

Start at tynker.com/modbedrock

- Double-check you typed the URL correctly! When you connect to a Tynker server for the first time, you should see a “Connection Successful” message.
- Try restarting Minecraft (close it, then open it again). This is especially useful when loading new Resource Packs (Addons).
- Make sure that you’ve triggered your code correctly. Double-check that you entered the correct command in chat or that the event actually occurred.

How Do I Buy Minecraft?

Minecraft Bedrock is the latest version of Minecraft, and you should use that if you are buying Minecraft for the first time. You’ll be able to use all of Tynker’s modding tools, as well as the latest and greatest interactive Tynker Minecraft courses. Minecraft Bedrock supports two platforms:

- **For Windows 10.** Working on a Windows PC? If you don’t already have a license, you can purchase and install **Minecraft for Windows 10** at **minecraft.net**—or from the Microsoft App Store.
- **Pocket Edition.** If you’re following along on an iPad, you can also use Minecraft Pocket Edition. Install it from the Apple App Store.

Tynker’s latest immersive Minecraft courses **Training Day**, **Agent Recall**, and **Gears of Time** are also compatible with **Minecraft Bedrock only** (Minecraft for Windows 10 or Pocket Edition).

No Connection?

You should see a “connection successful” message in chat appear after a successful **/connect** command. Double-check the web address given to you in Tynker and type it again, carefully. ***The most common error is misspelling the web address.***

Server Connected, But Nothing Happened?

This might be for a few reasons. If your code is triggered by a particular event, make sure that event is actually happening: Type in the exact message in chat, acquire the item or resource, or perform the action in question. Remember, you can enable creative mode by typing **/game-mode creative** in chat.

“Connection Closed.”

Any invalid code might make Tynker’s server return a message in chat, “Connection Closed.” If this happens, your code won’t run at all. You may need to test and isolate what is causing the problem. Try testing your program script by script, and if that doesn’t work, test your program line-by-line.

Cheats Disabled

In order to deploy mods, your world must allow “cheats.” Make sure to choose **Activate Cheats** when creating a Minecraft world to test your Tynker mods. To create a new world that allow cheats, start Minecraft then choose **Play > Create New > Create New World > Activate Cheats**.

Where's My Pack? Missing Resource Packs and Behavior Packs.

You must first download your resource pack from Tynker, then load it into Minecraft for it to become available. Try restarting your Minecraft client if you don't see it show up immediately in **My Packs**.

I want to see my item or block in-action!

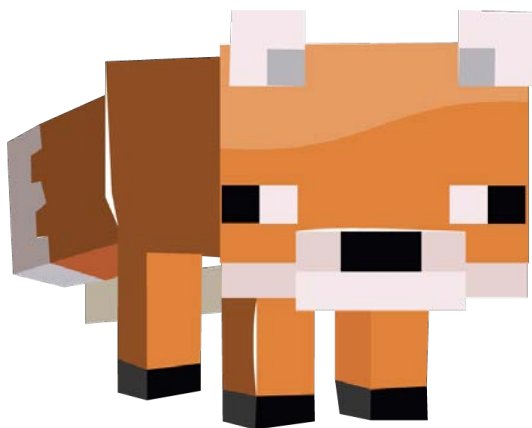
You can give yourself all items and blocks immediately, by enabling Creative Mode. Enter **/gamemode creative** in chat.

I want to Summon a mob!

Try using the command: **/summon panda** in chat, replacing "panda" with whichever mob you like.

How can I see my skin?

On Windows, press **F5** to cycle through different cameras (first person, over-the-shoulder, and so forth). On iPad, find camera controls in your Settings (**Pause > Settings > Video > Camera Perspective**).



Install Mod Creator

Find and install Tynker's Mod Creator in the app store.

Allow Access

When you run the Tynker Mod Creator iPad app for the first time, you'll be prompted to give the Tynker app access to your Minecraft folder on the iPad. If you allow access, you'll be able to deploy Minecraft mods, resource packs, and add-ons in a snap!

