

Teaching Kids to Code with Minecraft



Learning and having
fun.



Course Content

👉 Why Use Minecraft for Teaching Coding?

Minecraft: A Powerful Educational Tool

Minecraft is more than just a game—it's a learning platform that helps students develop:

- ✓ Problem-solving skills – Encourages logical thinking
 - ✓ Creativity – Students build and experiment in an open world
 - ✓ Programming fundamentals – Teaches coding through a hands-on approach
 - ✓ Engagement – Kids love Minecraft, making learning exciting and fun
- 👉 This course provides a structured approach to teaching coding, ensuring students grasp key concepts through theory, hands-on exercises, quizzes, and independent activities.

Course Content

👉 Why Use Minecraft for Teaching Coding?

Minecraft: A Powerful Educational Tool

Minecraft is more than just a game—it's a learning platform that helps students develop:

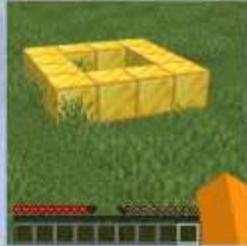
- ✓ Problem-solving skills – Encourages logical thinking
 - ✓ Creativity – Students build and experiment in an open world
 - ✓ Programming fundamentals – Teaches coding through a hands-on approach
 - ✓ Engagement – Kids love Minecraft, making learning exciting and fun
- 👉 This course provides a structured approach to teaching coding, ensuring students grasp key concepts through theory, hands-on exercises, quizzes, and independent activities.

Course Information

Quick Start 1



Let's have an immediate result!



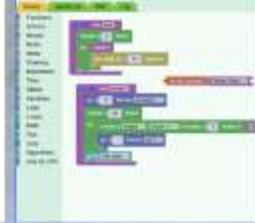
The Coding Editor 2

Visualmodder



A quick overview of the coding editor

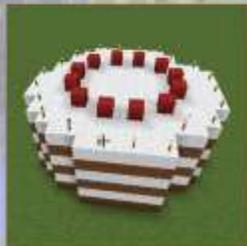
Visualmodder



Iteration with Simple Loops 3



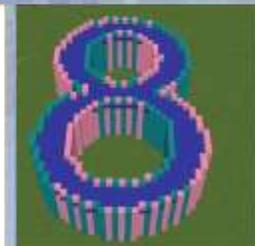
Learn to use the basic repeat command



Combining Blocks 4



Create beautiful structures by combining blocks



Course Information

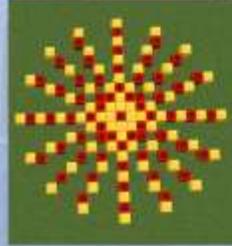
Moving in the world 5



Learn how to control the robot position



Horizontal Rotation 6



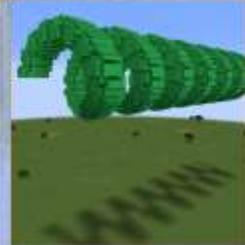
Amazing structures created with simple rotations



Vertical Rotation 7



Amazing structures created with simple rotations



Functions 8



Organize code into functions



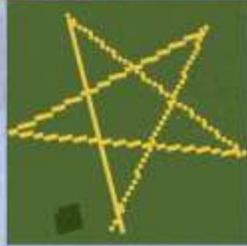
Course Information

Advanced Positioning

9

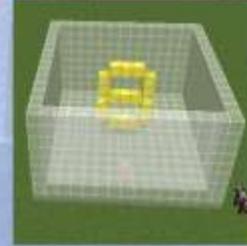


Learn how to mark specific positions

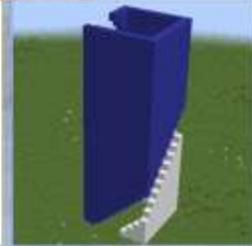


Variables

10

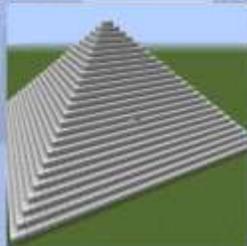


Understand what variables are and why we need them

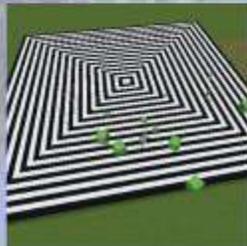


Counting Loops

11



Learn to use the "for" loop



Logic and Conditionals

12

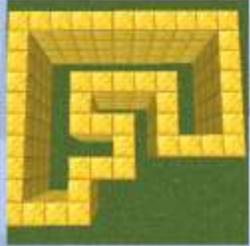


"If .. then .. Else" and random numbers



Course Information

Complex Shapes 13



Create non-geometric shapes



Conclusion Part 1 8

What we learned so far

Part 2 (to be published) 10

Advanced blocks
While loops
Events
Parameters

Part 3 (to be published) 11

Lists
Return values
Debugging
Coordinate systems

What You Need to Get Started

- ✓ Minecraft Java Edition – Each student needs a license if they don't already own it
 - ✓ Access to the VisualModder online server – No software installation required
 - ✓ 1 to 3 students per session – Ideal for small-group teaching
 - ✓ A basic understanding of computers – No prior coding knowledge necessary
- 💡 With these simple requirements, teachers can immediately start delivering coding lessons in an interactive and engaging way.

How VisualModder Works

Simplifying Coding with a Visual Interface

◆ What is VisualModder? A online editor that allows students to code using a block-based, drag-and-drop system within Minecraft.

Why use it?

- ✓ Removes the complexity of syntax errors with block coding
- ✓ Provides instant feedback within Minecraft
- ✓ Teaches fundamental coding logic in an accessible way
- ✚ Students will start with simple commands and gradually move toward more advanced coding projects, all within the Minecraft world!
- ◆ If you have a maximum of 3 students and don't need a personal server, you can use the server at www.visualmodder.org at no cost. Otherwise you can download the free plugin and deploy it on your own Minecraft Server.

Lesson types

Step-by-Step Learning Path

 Theory Introduction – Slides explain coding concepts in a simple and visual way

 Guided Exercises – Students follow along and build their first programs

 Quizzes – Quick tests to reinforce understanding

 Independent Challenges – Open-ended projects to encourage creativity

 By the end of the course, students will have created and run their first Minecraft program, building a customizable tower with different block types.

Start Teaching Today!

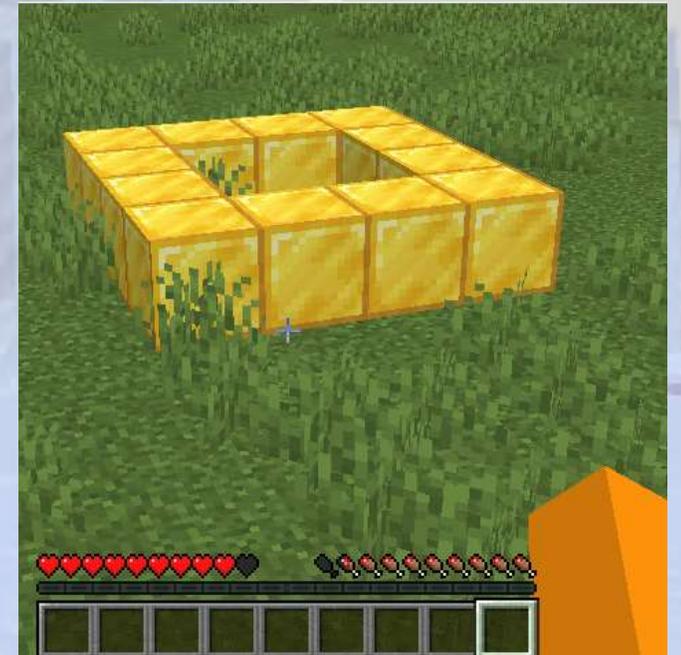
Empower Students Through Coding

-  Follow the slides to introduce coding concepts gradually
-  Use VisualModder for hands-on, interactive learning
-  Encourage students to experiment and explore beyond the exercises
-  Leverage quizzes and independent projects to solidify learning
-  This PowerPoint is designed to be a complete teaching tool, providing everything needed to guide students from beginners to confident coders within Minecraft.
-  Ready to begin? Let's dive into the first lesson!

Quick Start



Let's have an
immediate result!



Quick Start

Section Overview

This section introduces students to the fundamental tools, servers, and basic commands necessary for coding in Minecraft.

Objectives

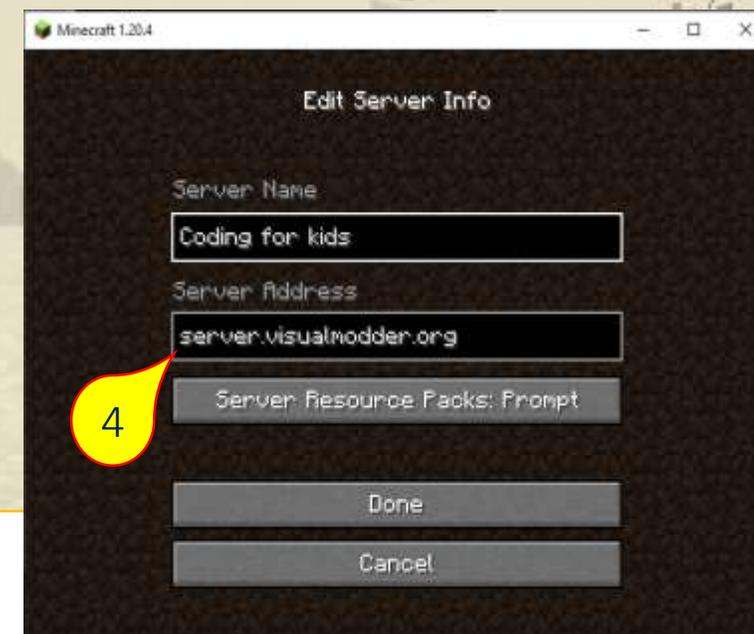
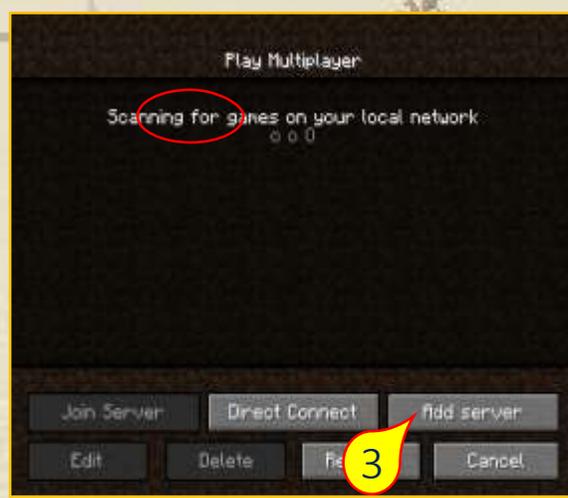
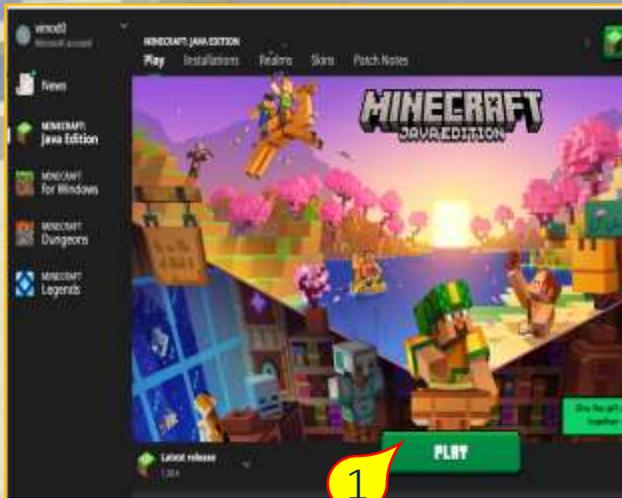
The main goal is to ignite students' interest in coding and provide them with an early sense of achievement, encouraging a passion for programming.

Expected Outcomes

By the end of this section, students will have successfully created and executed their first Minecraft program, which generates a customizable tower using different block types.

⚡ Step 1: Connect to the Minecraft Server

Step-by-step guide to joining the Minecraft server



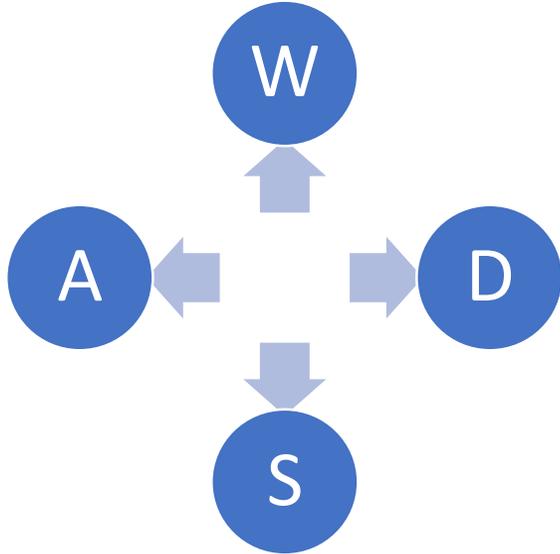
1. Start Java Minecraft
2. Start the game in “multiplayer” mode
3. Add a new server
4. Enter “server.visualmodder.org” and press done
5. Now you can join the server and start playing

⚡ Step 1: Connect to the Minecraft Server

Now you should be able to move around in the Minecraft world.

Here are some basics:

Movement



Additional keys:

Jump: press space bar

Fly: double click space bar

Inventory: click letter E

Commands: click symbol "/" or "-"

⚡ Step 2: Create your first program

Open the visualmodder.org webpage and click on the “CODE EDITOR” button

www.visualmodder.org

VisualModder.org

HOME HOSTING PLUGIN TUTORIALS ARTICLES ABOUT US CONTACT

Teach coding with Minecraft Java

Free, no registration needed

Welcome to our free coding server, designed for teaching coding to children and beginners using Minecraft java edition. Utilizing block coding, we offer a user-friendly experience to establish a robust programming foundation.

Accessible through the regular Java Minecraft game, our solution has empowered over 3000 students worldwide to acquire programming skills.

Try it now! Simply add the server "server.visualmodder.org" to your java Minecraft game and click the button below to access the code editor for creating your own programs.

CODE EDITOR

Examples

| | | | |
|-----------|---------------|--------------|--------------|
| Sphere | Tower | Slide | Eiffel tower |
| Medusa | Train | SpaceShip | City |
| Labyrinth | Door | Bridge | Orientation |
| Stadium | Motorboat | Diaco | Sailing Boat |
| Skull | Swimming Pool | Tennis Field | Coca Cola |

Visualmodder

Click here to support us with a donation ..

Minecraft player name: English

Examples ...

Blocks JavaScript XML Log

- Functions
- Actions
- Objects
- Time
- Movement
- Drawing
- Events
- Variables
- Logic
- Loops
- Math
- Text
- Lists
- Algorithms
- Op only

⚡ Step 2: Create your first program

This is the empty page of the program editor.

To understand it better, Imagine that an invisible robot is working for us and this empty page is it's brain. We have to add programs so that it knows what we want it to do.



⚡ Step 2: Create your first program

First we have to put our player name in the field indicated, otherwise we will not find our programs in Minecraft

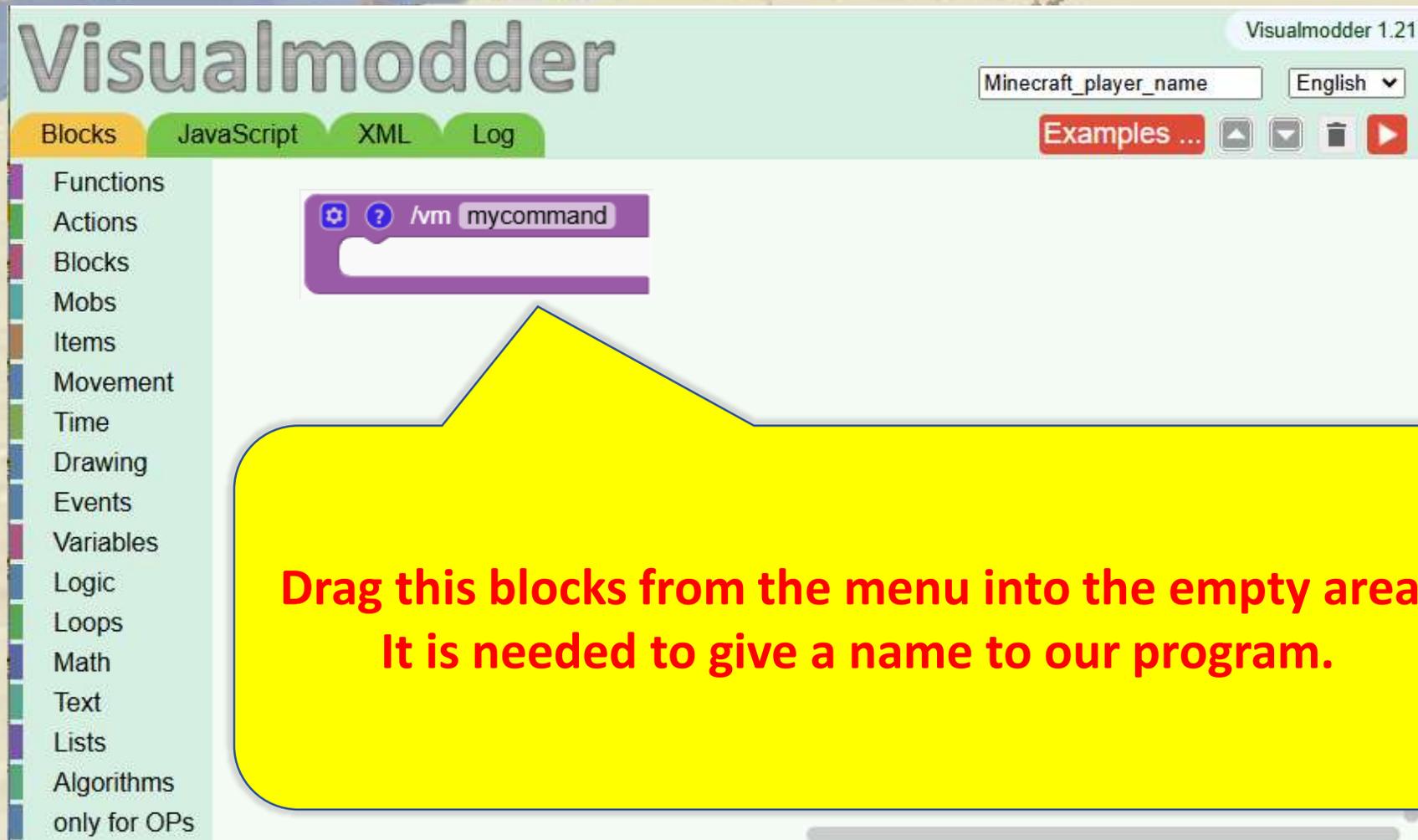


The screenshot shows the Visualmodder 1.21 interface. At the top, the title "Visualmodder" is displayed in a large, grey, textured font. To the right of the title, the version "Visualmodder 1.21" is shown. Below the title, there are four tabs: "Blocks", "JavaScript", "XML", and "Log". The "JavaScript" tab is currently selected. On the left side, there is a vertical menu with various categories: Functions, Actions, Blocks, Mobs, Items, Movement, Time, Drawing, Events, Variables, Logic, Loops, Math, Text, Lists, Algorithms, and only for OPs. In the top right corner, there is a text input field containing "Minecraft_player_name" and a dropdown menu set to "English". Below these, there is a red button labeled "Examples ..." and several small icons. A yellow callout box with a blue border points to the "Minecraft_player_name" input field. Inside the callout box, the text reads: "Put here the player name you are using in Minecraft. It is the one you picked when you bought Minecraft".

Put here the player name you are using in Minecraft. It is the one you picked when you bought Minecraft

⚡ Step 2: Create your first program

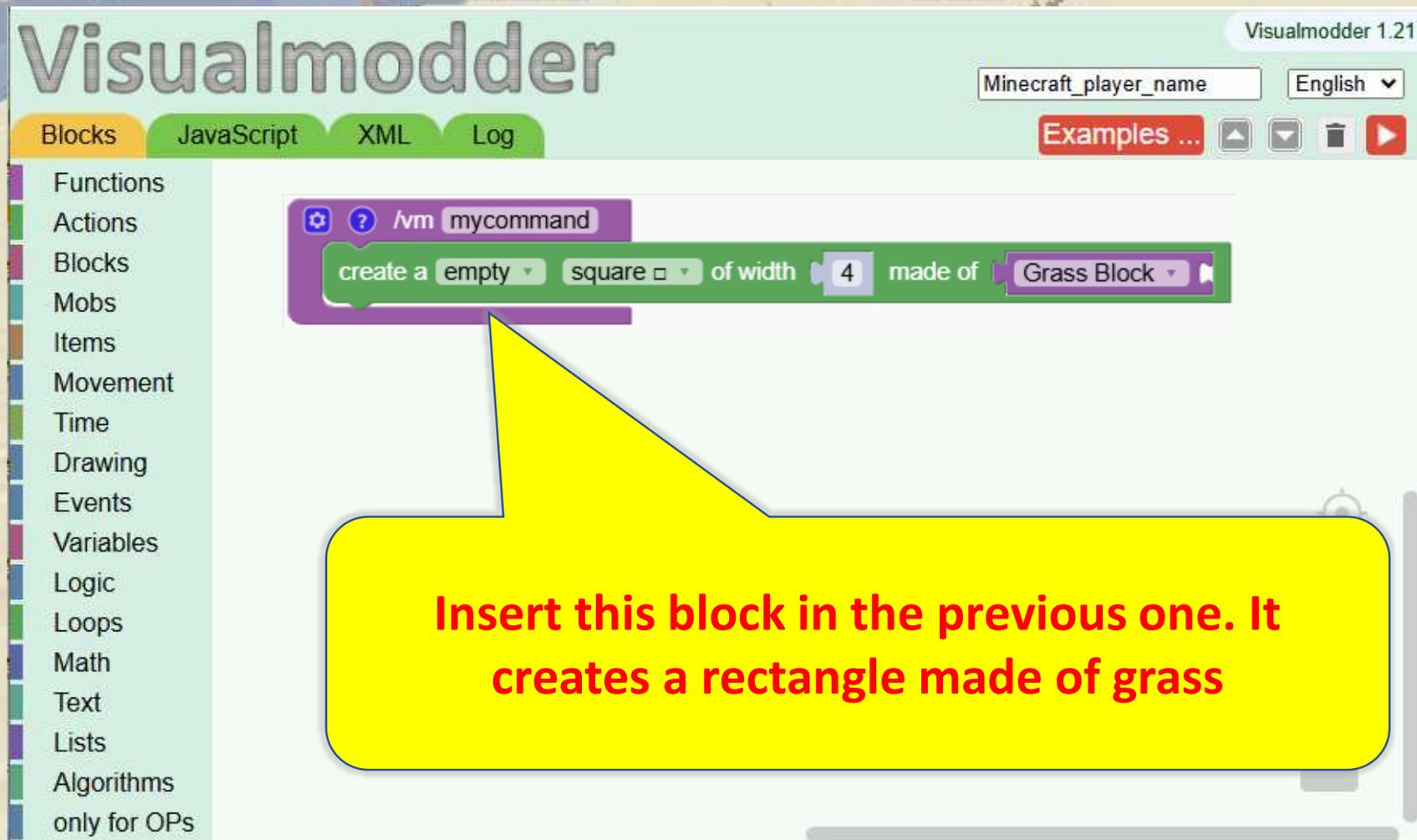
We create a first simple program that builds a square made of blocks.



The screenshot shows the Visualmodder 1.21 interface. At the top, there's a header with the title "Visualmodder" and a version number "Visualmodder 1.21". Below the header, there are tabs for "Blocks", "JavaScript", "XML", and "Log". The "JavaScript" tab is selected. On the right side, there are input fields for "Minecraft_player_name" and "English", along with an "Examples ..." button and some navigation icons. A central workspace contains a single purple block with a gear icon, a question mark, and the text "/vm mycommand". A yellow callout box with red text points to this block, stating: "Drag this blocks from the menu into the empty area. It is needed to give a name to our program." On the left side, there is a vertical menu with various categories: Functions, Actions, Blocks, Mobs, Items, Movement, Time, Drawing, Events, Variables, Logic, Loops, Math, Text, Lists, Algorithms, and only for OPs.

⚡ Step 2: Create your first program

Block coding allows to connect code like puzzle pieces



The screenshot shows the Visualmodder 1.21 interface. At the top, there's a header with the name "Visualmodder" and a version number "Visualmodder 1.21". Below the header, there are tabs for "Blocks", "JavaScript", "XML", and "Log". On the right side, there are input fields for "Minecraft_player_name" and "English", along with an "Examples ..." button and some navigation icons. A left sidebar lists various categories: Functions, Actions, Blocks, Mobs, Items, Movement, Time, Drawing, Events, Variables, Logic, Loops, Math, Text, Lists, Algorithms, and only for OPs. The main workspace displays a block coding script. The first block is a purple block labeled "/vm mycommand". Attached to its bottom is a green block that says "create a empty square of width 4 made of Grass Block". A yellow callout box with red text points to the green block, stating: "Insert this block in the previous one. It creates a rectangle made of grass".

⚡ Step 2: Create your first program

We pick gold and now the program is ready.



The screenshot shows the Visualmodder 1.21 interface. The title bar includes the name 'Visualmodder' and the version 'Visualmodder 1.21'. Below the title bar, there are tabs for 'Blocks', 'JavaScript', 'XML', and 'Log'. A search bar contains 'Minecraft_player_name' and a language dropdown is set to 'English'. A red 'Examples ...' button is visible. On the left, a vertical menu lists various categories: Functions, Actions, Blocks, Mobs, Items, Movement, Time, Drawing, Events, Variables, Logic, Loops, Math, Text, Lists, Algorithms, and 'only for OPs'. The main workspace contains a script with a purple block labeled '/vm mycommand' and a green block labeled 'create a empty square of width 4 made of Block of Gold'. A yellow callout box with a red border points to the 'Block of Gold' dropdown menu, containing the text 'Change the material to gold'. On the right side of the workspace, there are icons for zooming in (+), zooming out (-), and a trash can.

⚡ Step 3: Run your first program in Minecraft

We go back to Minecraft and with the command key '/' we type 'vm mycommand' which tells our robot to run the program called 'mycommand'



⚡ Step 3: Run your first program in Minecraft

You did it! You ran your first program in Minecraft.



⚡ Step 4: Let's do it again with a tower

Let's repeat the process, but this time we'll make a tower



⚡ Step 4: Let's do it again with a tower

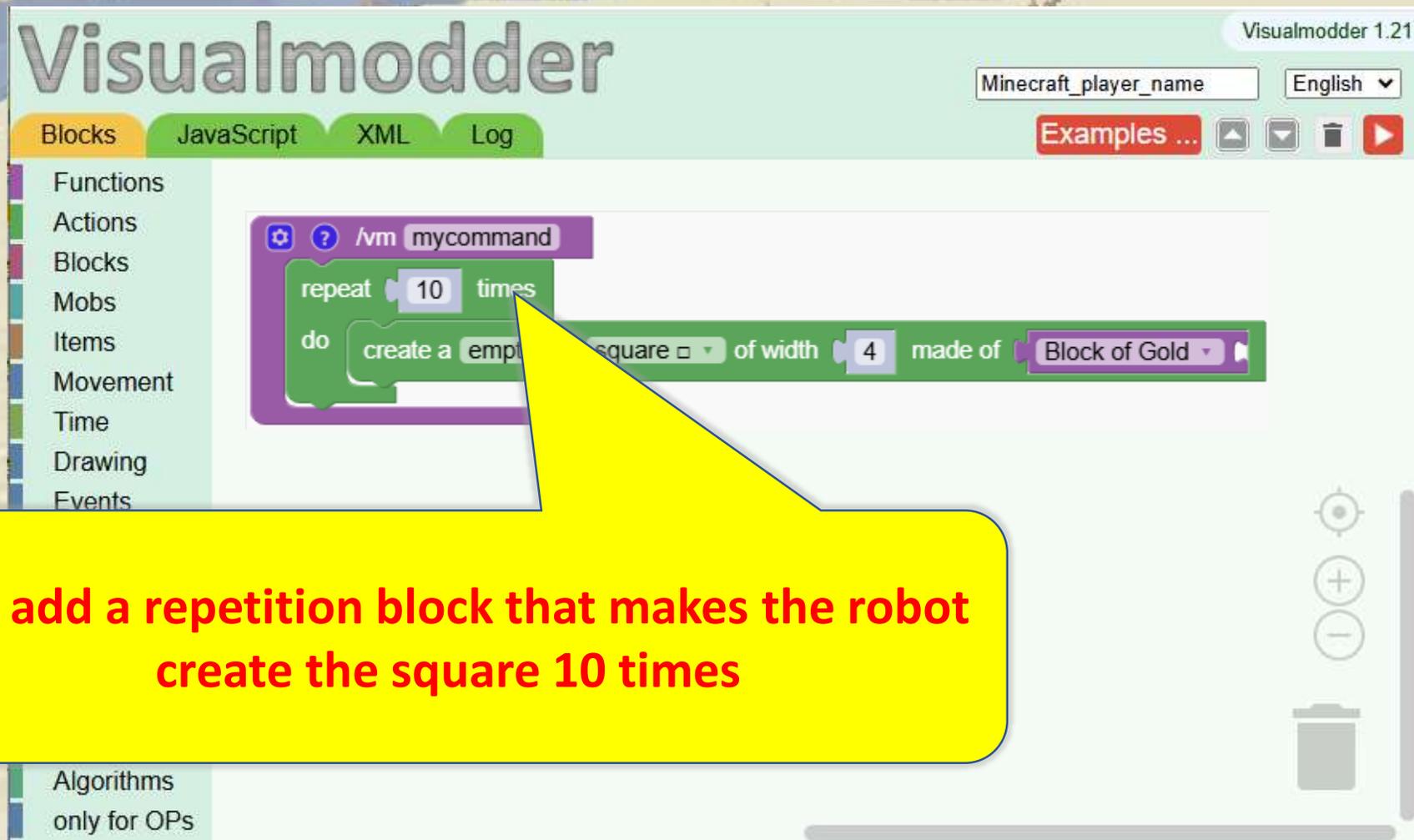
We modify the previous program.



The screenshot shows the Visualmodder 1.21 interface. The title bar includes the name 'Visualmodder' and the version 'Visualmodder 1.21'. Below the title bar, there are tabs for 'Blocks', 'JavaScript', 'XML', and 'Log'. On the right side of the title bar, there is a text input field containing 'Minecraft_player_name', a language dropdown menu set to 'English', and a red 'Examples ...' button with navigation icons. A sidebar on the left lists various categories: Functions, Actions, Blocks, Mobs, Items, Movement, Time, Drawing, Events, Variables, Logic, Loops, Math, Text, Lists, Algorithms, and 'only for OPs'. The main workspace contains a single code block with the following text: `/vm mycommand` followed by a green block containing `create a empty square of width 4 made of Block of Gold`. A yellow callout box with a red border points to the code block and contains the text: **We modify the program to make a tower**. On the right side of the workspace, there are zoom controls (a target icon, a plus sign, a minus sign, and a trash can icon).

⚡ Step 4: Let's do it again with a tower

We repeat the square 10 times



Visualmodder 1.21

Minecraft_player_name English

Examples ...

Blocks JavaScript XML Log

Functions
Actions
Blocks
Mobs
Items
Movement
Time
Drawing
Events

Algorithms
only for OPs

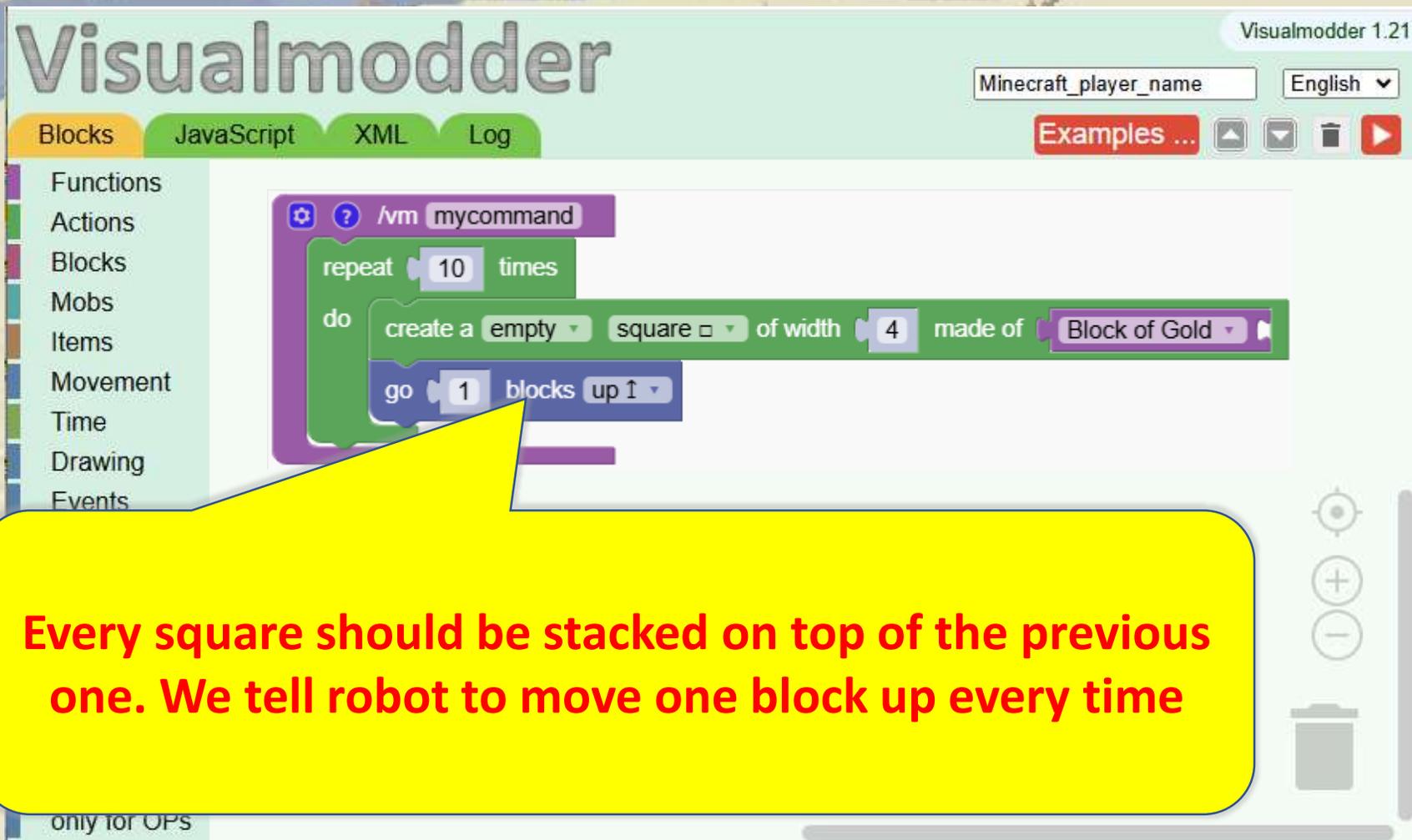
repeat 10 times

do create a empty square of width 4 made of Block of Gold

We add a repetition block that makes the robot create the square 10 times

⚡ Step 4: Let's do it again with a tower

Robots need to be repositioned after creating a square



Visualmodder 1.21

Minecraft_player_name English

Examples ...

Blocks JavaScript XML Log

Functions
Actions
Blocks
Mobs
Items
Movement
Time
Drawing
Events

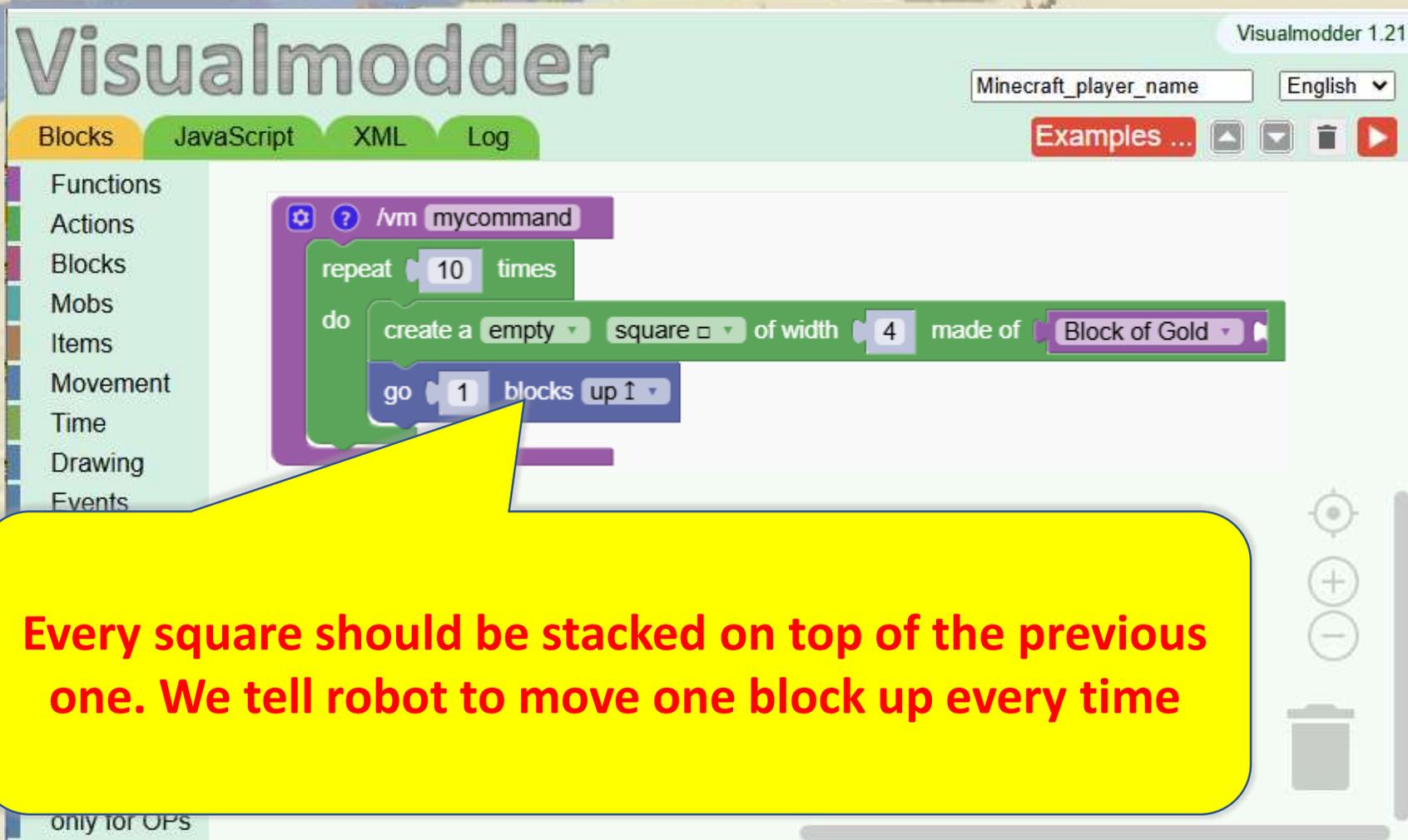
```
repeat 10 times
do
  create a empty square of width 4 made of Block of Gold
  go 1 blocks up
```

Every square should be stacked on top of the previous one. We tell robot to move one block up every time

only for OPS

⚡ Step 4: Let's do it again with a tower

The program is ready.



The screenshot shows the Visualmodder 1.21 interface. The main workspace contains a script with the following blocks:

- `/vm mycommand`
- `repeat 10 times`
- `do` block containing:
 - `create a empty square of width 4 made of Block of Gold`
 - `go 1 blocks up`

A yellow callout box with red text explains the logic: "Every square should be stacked on top of the previous one. We tell robot to move one block up every time".

⚡ Step 4: Let's do it again with a tower

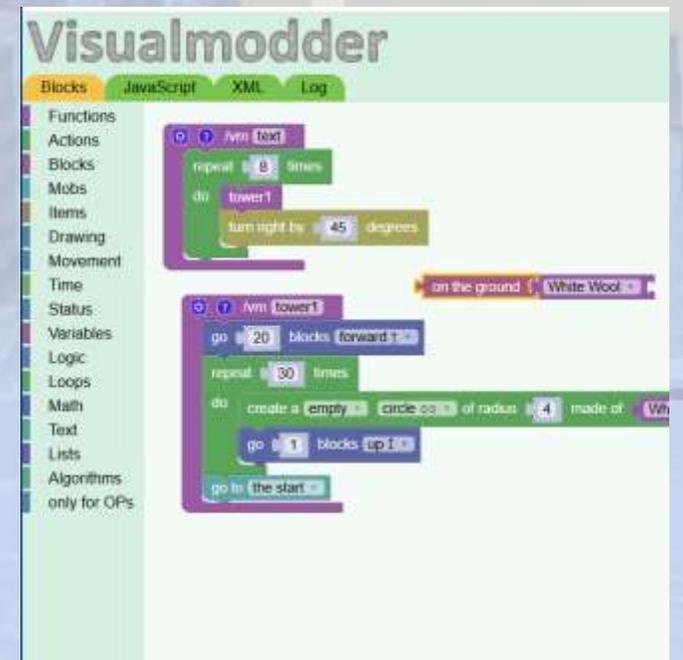
Type 'vm mycommand' and the tower will appear. (We are inside it. Just fly out 😊)



The Coding Editor



A quick overview of
the coding editor



The Coding Editor

Section Overview

We explore the features and user interface of the coding tool to manage and edit programs efficiently.

Objectives

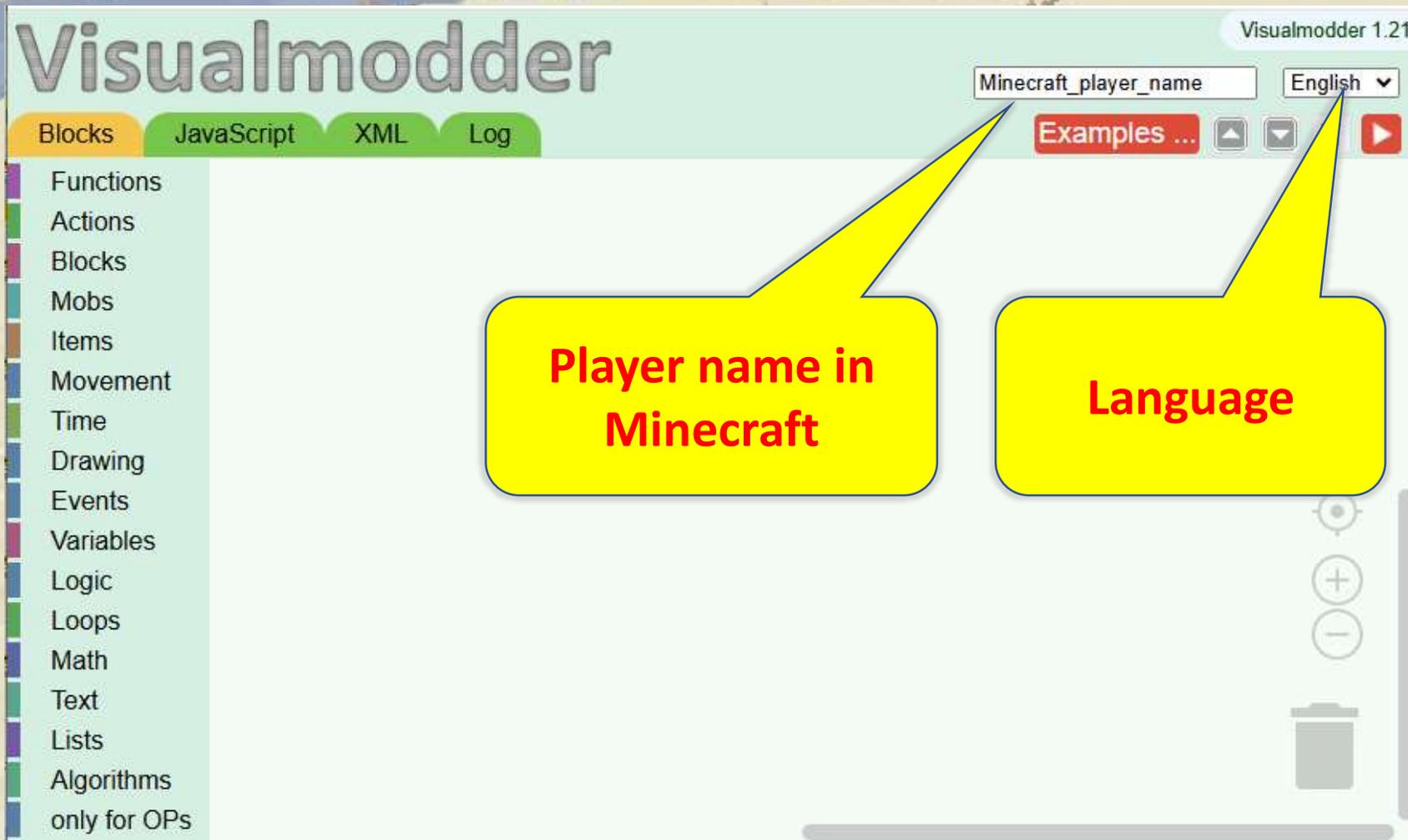
The main goal is to make the coding effort as easy as possible to keep the student's focused on the coding itself

Expected Outcomes

Understand the essential functions of the coding editor, such as saving, reloading, and organizing programs for better workflow.

Explanation of the Editor User Interface

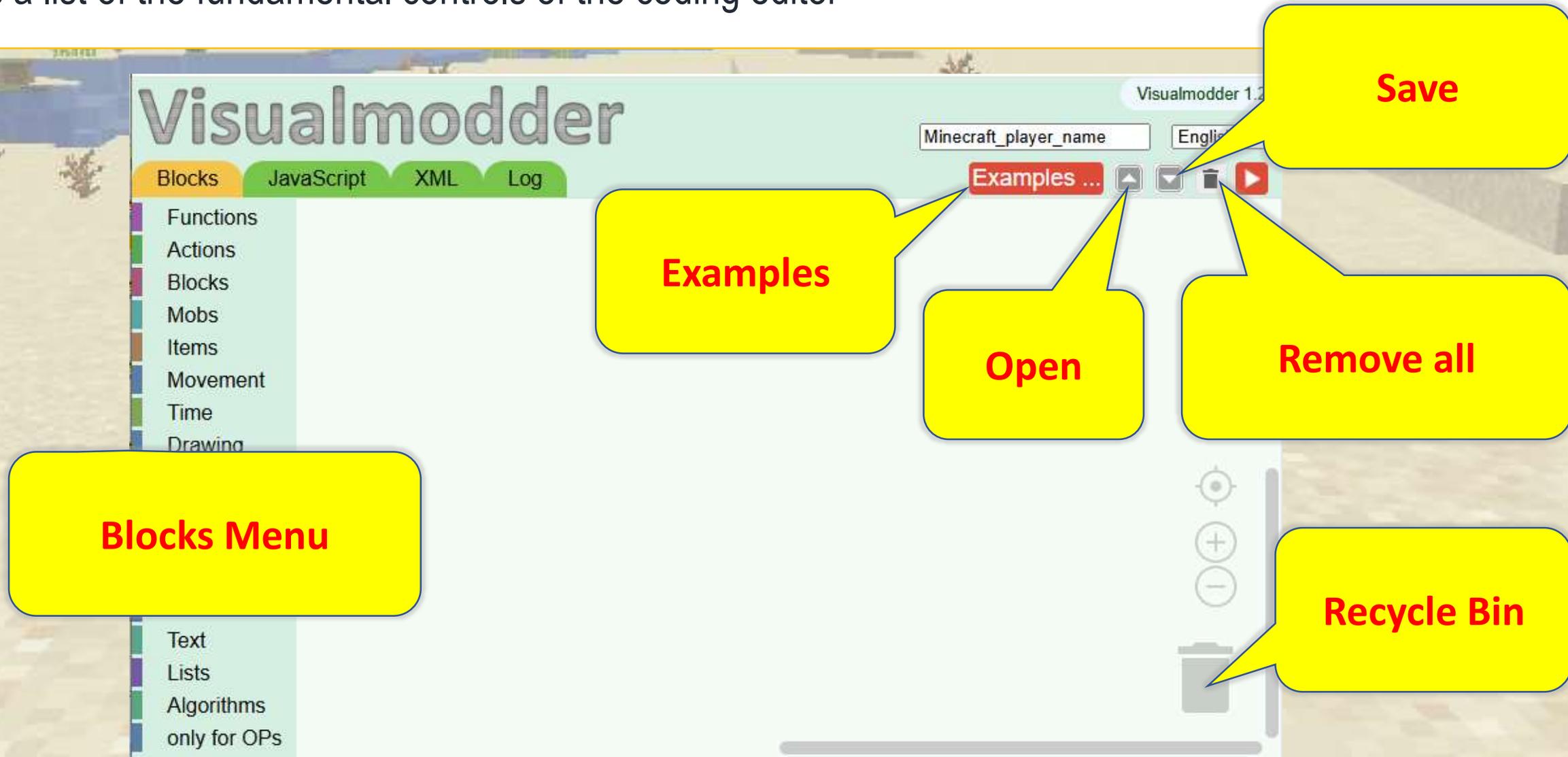
Set your preferred language and enter the player name you are using when playing in Minecraft.



The screenshot shows the Visualmodder 1.21 editor interface. The title bar displays "Visualmodder 1.21". Below the title bar, there are tabs for "Blocks", "JavaScript", "XML", and "Log". The "JavaScript" tab is currently selected. On the right side of the title bar, there is a text input field containing "Minecraft_player_name" and a dropdown menu set to "English". Below these fields is a red button labeled "Examples ..." with up and down arrow icons and a play button icon. A yellow callout box points to the "Minecraft_player_name" input field with the text "Player name in Minecraft". Another yellow callout box points to the "English" dropdown menu with the text "Language". On the left side, there is a vertical sidebar with a list of categories: Functions, Actions, Blocks, Mobs, Items, Movement, Time, Drawing, Events, Variables, Logic, Loops, Math, Text, Lists, Algorithms, and only for OPs. At the bottom right, there are several icons: a gear, a plus sign, a minus sign, and a trash can.

Features of the Coding Editor

Here is a list of the fundamental controls of the coding editor



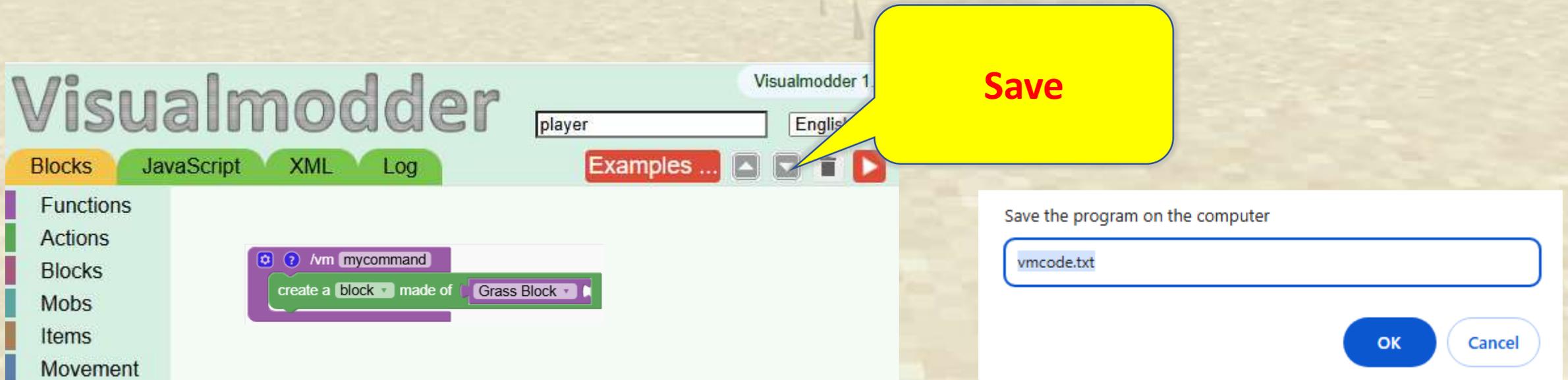
Save and Reload a Program

Practice saving, reloading, and continuing your work without losing progress.

Step 1:

Create some blocks and then click on the save button.

In the popup window you can choose the name of the file to create on your computer



The screenshot shows the Visualmodder interface. At the top, the title "Visualmodder" is displayed. Below it, there are tabs for "Blocks", "JavaScript", "XML", and "Log". A search bar contains the text "player" and a language dropdown is set to "English". A red button labeled "Examples ..." is visible. A yellow callout bubble with the word "Save" in red points to a save icon in the interface. In the center, a code block is visible with the text: `/vm mycommand` followed by a block "create a block made of" containing a "Grass Block". On the right, a "Save the program on the computer" dialog box is open, showing a text input field with "vmcode.txt" and "OK" and "Cancel" buttons.

Save and Reload a Program

Practice saving, reloading, and continuing your work without losing progress.

Step 2:
Clean the workspace

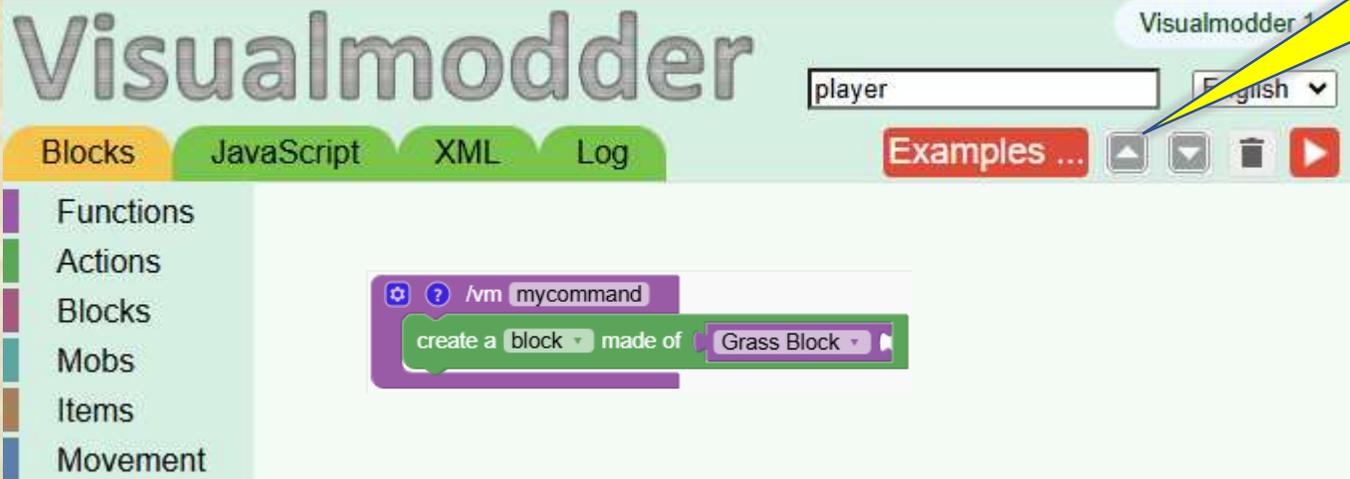


Remove all

Save and Reload a Program

Practice saving, reloading, and continuing your work without losing progress.

Step 3:
Reload your file. The workspace should now contain the same blocks that you saved in Step 1

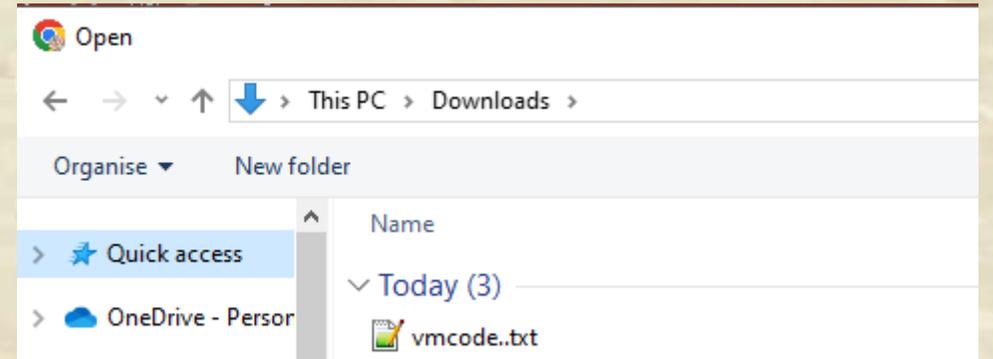


The screenshot shows the Visualmodder interface. The title bar reads "Visualmodder 1". Below the title bar, there is a search bar containing the text "player" and a language dropdown menu set to "English". A red button labeled "Examples ..." is visible. The main workspace contains a code block with the following structure:

```
vm mycommand  
  create a block made of Grass Block
```

The left sidebar lists various categories: Functions, Actions, Blocks, Mobs, Items, and Movement.

Open



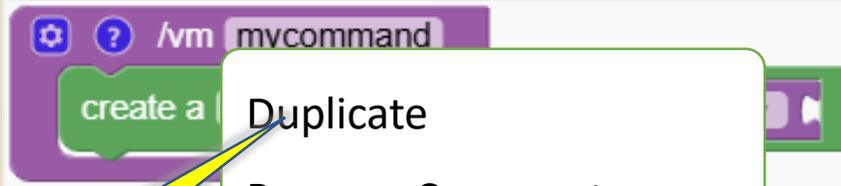
The screenshot shows a Windows File Explorer window titled "Open". The address bar indicates the current location is "This PC > Downloads >". The "Organise" menu is open, showing options like "New folder". A list of files is displayed, including a file named "vmcode..txt" under the "Today (3)" group.



Organizing Code

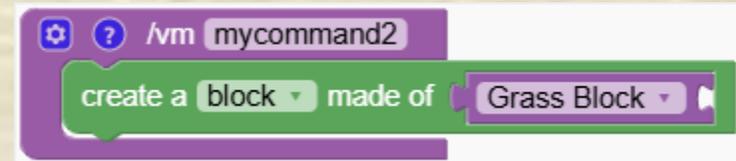
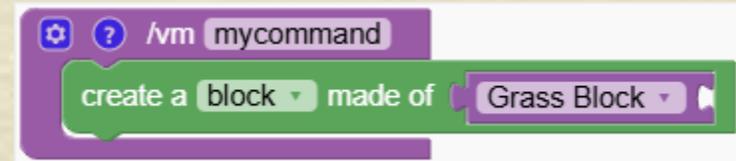
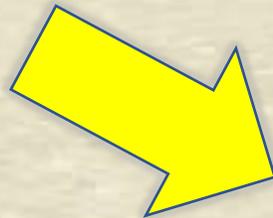
Right-click menu has many useful operations:

Duplicate a block



- Duplicate
- Remove Comment
- Collapse Block
- Disable Block
- Delete 3 Blocks
- Help
- Create 'mycommand'
- Download Image

duplicate

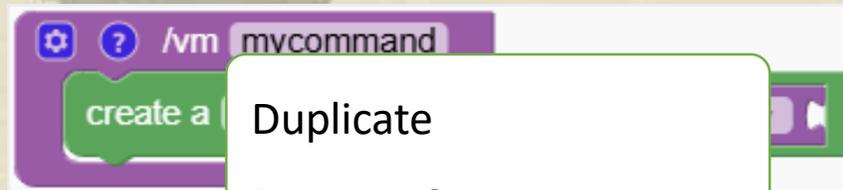


Organizing Code

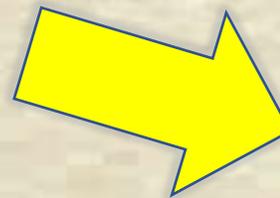
Right-click menu has many useful operations:

With the operation 'collapse' it is possible to shrink a block to save space

Once collapsed, in the menu we find the operation 'expand'



Collapse



`/vm mycommand create a emp...`

Running programs in Minecraft

In order to run a program in minecraft you use the command 'vm'

To access the command dialog you have to press the '/' character. This is configurable in the options of Minecraft.

 The 't' character opens a different dialog for chatting

\vm mycommand ► (executes the program called 'mycommand')

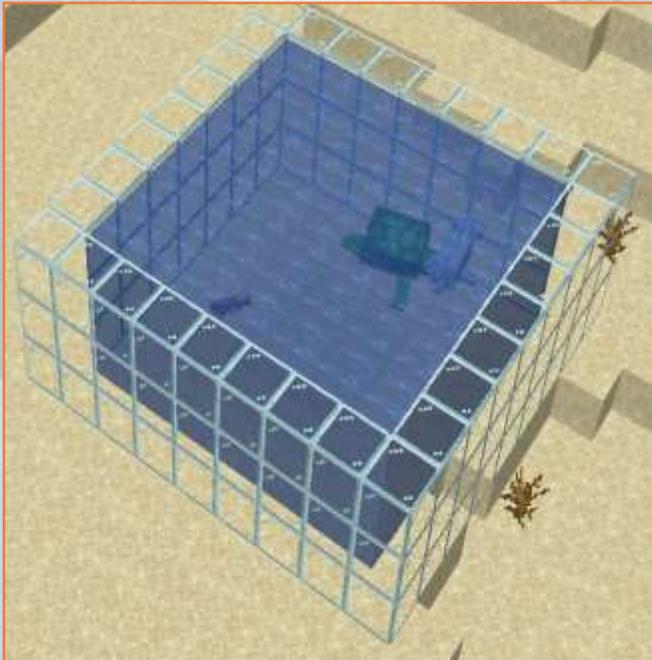
\vmu ► (Undoes the last creation)

\vmu mycommand ► (Undoes the last creation, and the runs the program 'mycommand' again)

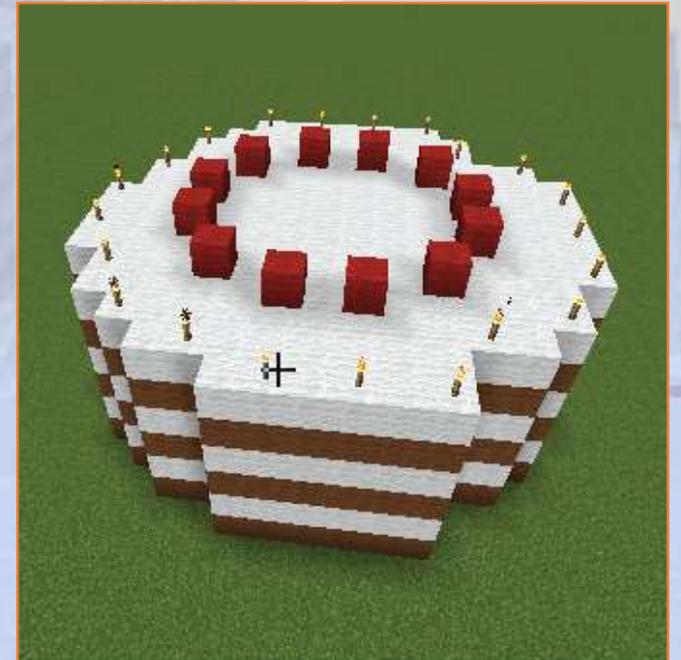
\vmtp 100 100 ► (Teleport to coordinates 100, 100)



Iteration with Simple Loops



Learn to use the
basic repeat
command



Iteration with Simple Loops

Section Overview

We follow guided exercises that explain how to create some amazing structures with the use of simple loops.

Objectives

Amaze kids with the power of coding. They learn that coding makes them more efficient in creating big structures

Expected Outcomes

Understand how loops can simplify repetitive tasks and easily create designs like towers, cakes, and any repetitive structures.

⚡ Let's Create an Aquarium

Combine mobs and blocks to create a beautiful aquarium.



⚡ Let's Create an Aquarium

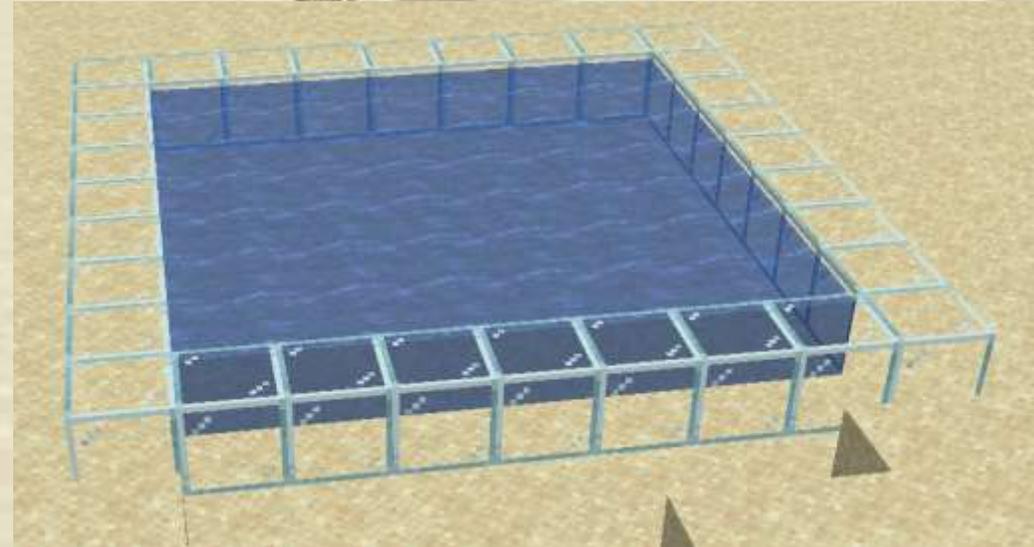
First, we create a square with a side of width 9 blocks, made of glass

Second, we fill the square it with a smaller square made of water

```

? /vm aquarium
create a empty square of width 9 made of Glass
create a empty square of width 7 made of Water

```



⚡ Let's Create an Aquarium

We repeat the process 3 times moving up one block every time

```
vm aquarium
repeat 3 times
do
  create a empty square of width 9 made of Glass
  create a empty square of width 7 made of Water
  go 1 block up
```



⚡ Let's Create an Aquarium

Now we add 3 different mobs.

```
vm aquarium
repeat 3 times
do
  create a empty square of width 9 made of Glass
  create a empty square of width 7 made of Water
  go 1 block up
create a block made of Dolphin
create a block made of Turtle
create a block made of Cod
```



⚡ Be prepared for survival

Using loops to improve our equipment



⚡ Be prepared for survival

Using loops to improve our equipment

In the blocks menu there are 3 interesting blocks for upgrading our equipment:

Load in inventory

Wear an armor

Yield a tool



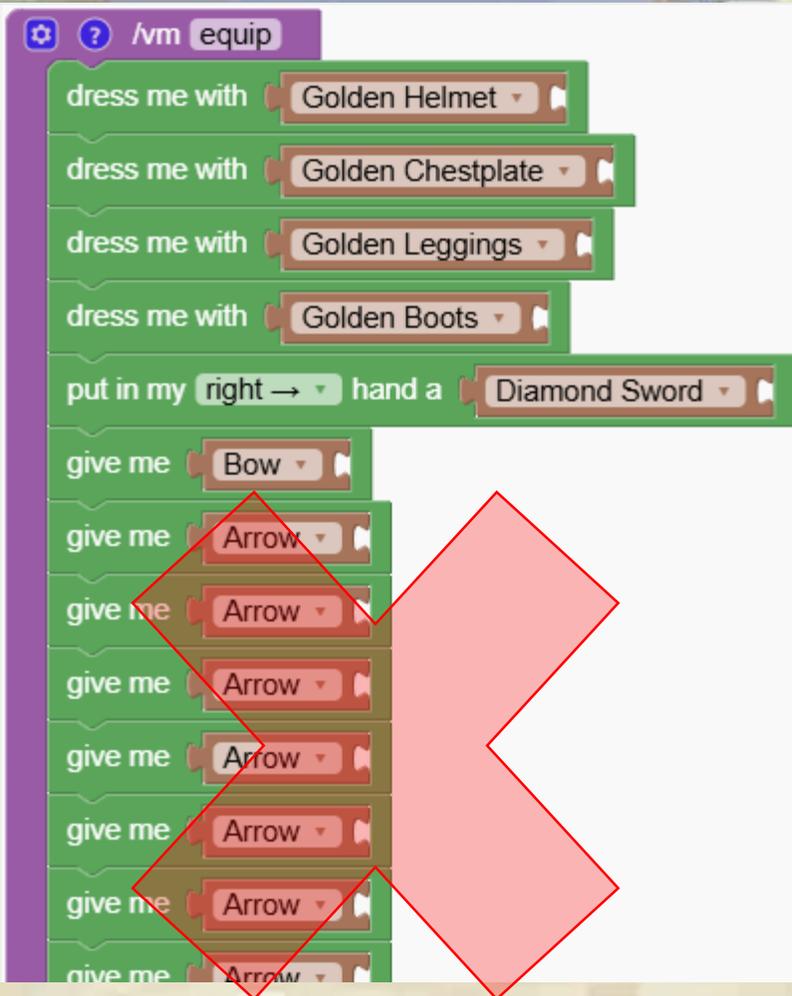
⚡ Be prepared for survival

Now we are getting all the gear but we have just one arrow.



⚡ Be prepared for survival

Repeating the block 64 times is a bad idea



⚡ Be prepared for survival

With one simple loop we get 64 arrows

```
/? /vm equip  
dress me with Golden Helmet  
dress me with Golden Chestplate  
dress me with Golden Leggings  
dress me with Golden Boots  
put in my right hand a Diamond Sword  
give me Bow  
repeat 64 times  
do give me Arrow
```



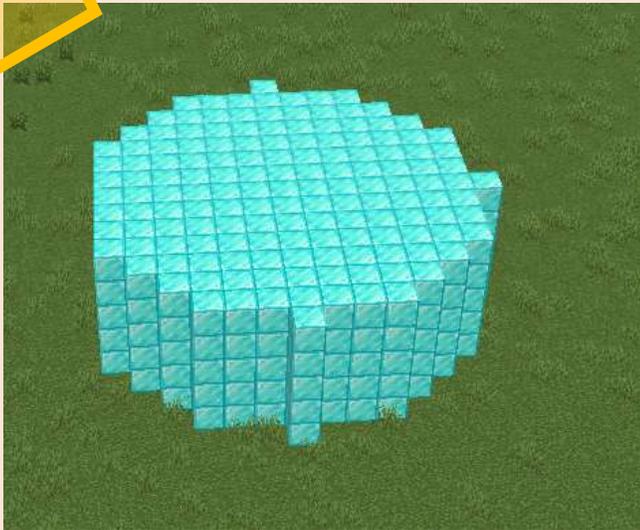


What does this program create?

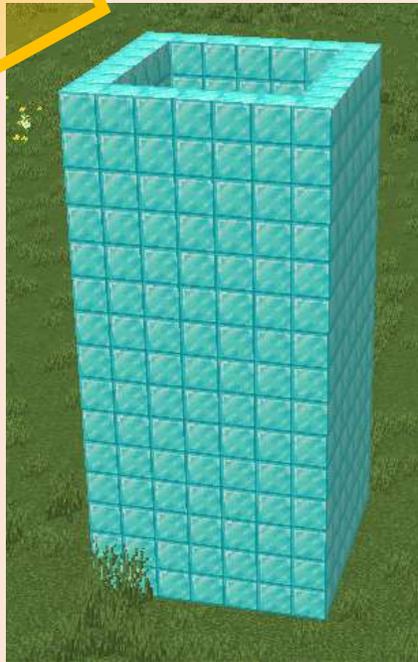
A, B or C?

```
repeat 5 times  
do  
  create a empty circle of radius 7 made of Block of Diamond  
  go 1 blocks up 1
```

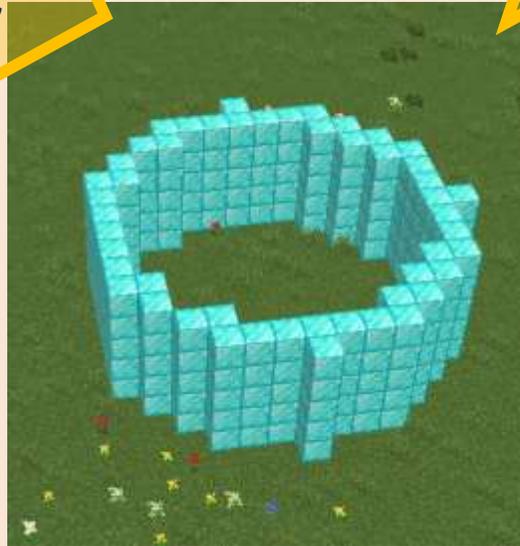
A



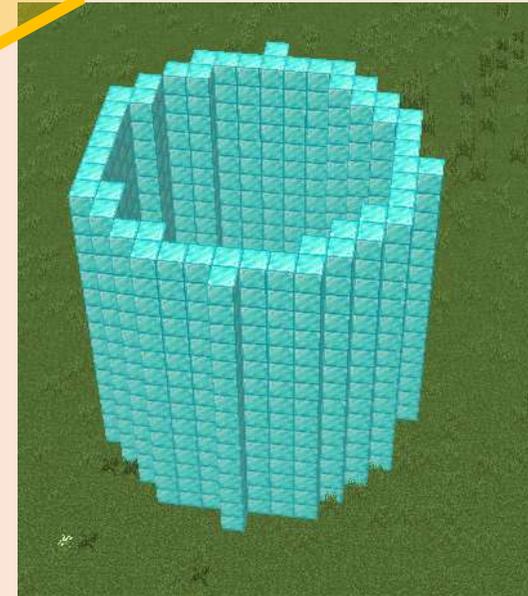
B



C



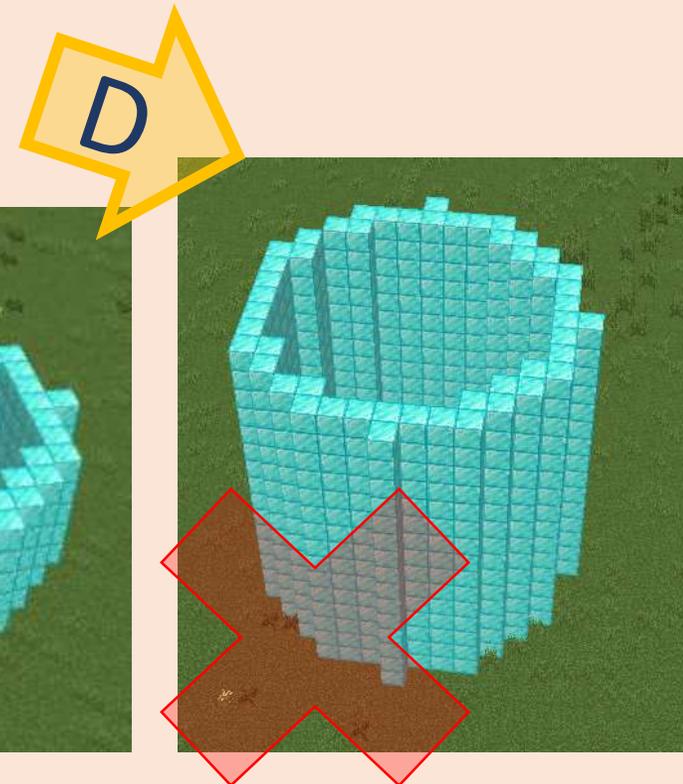
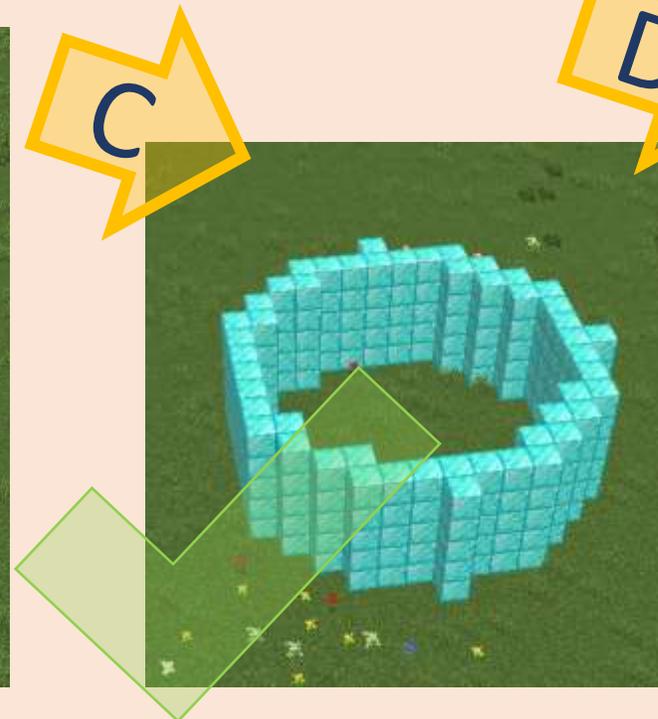
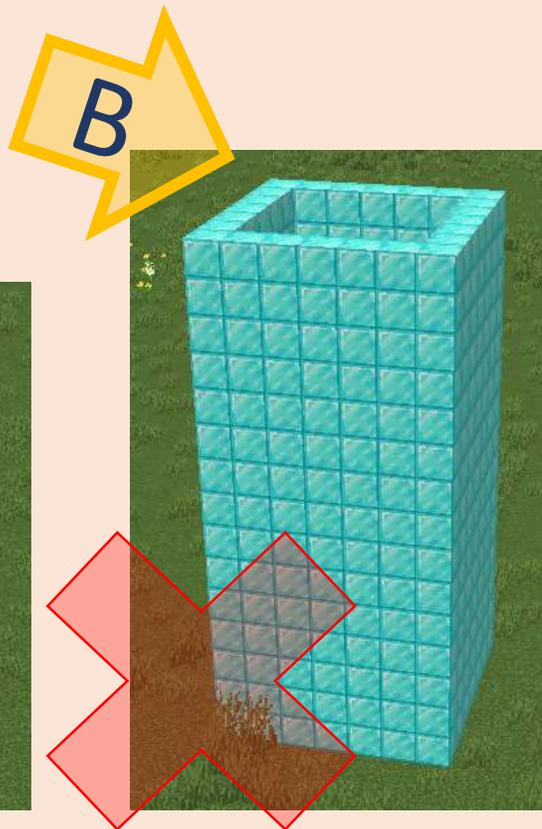
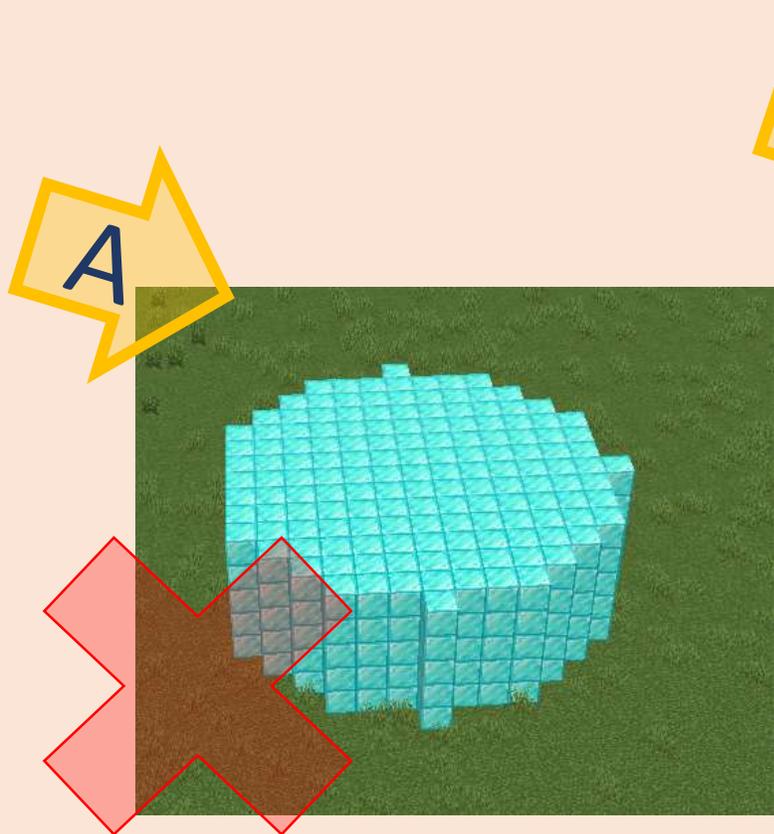
D



What does this program create?

A, B or C?

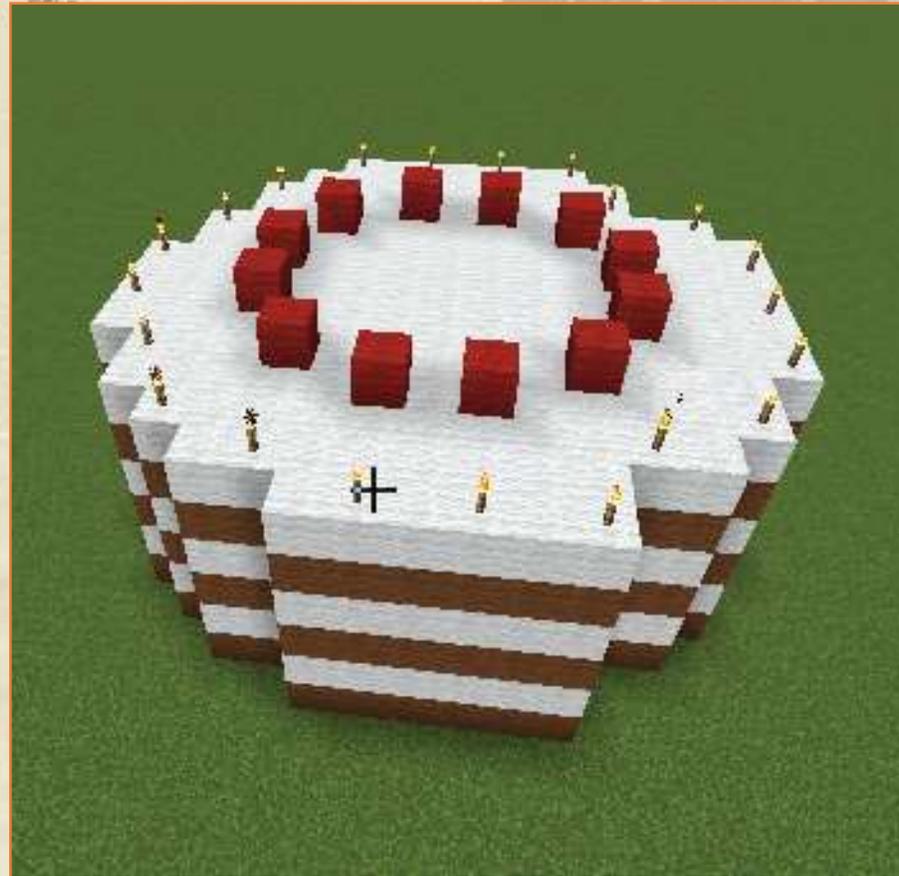
```
repeat 5 times
do
  create a empty circle of radius 7 made of Block of Diamond
  go 1 blocks up 1
```



⚡ Create a Birthday Cake

We are using loops to design and build a cake in Minecraft.

Our cake features layers inspired by vanilla and chocolate, topped with cherries and adorned with candles.



⚡ Create a Birthday Cake



First, we move the start position 10 steps further away to avoid being trapped in the cake.

Second, we create a simple circular tower. With a radius of 7 blocks

```
/vm cake
go 10 block forward ↑
repeat 3 times
do
  create a empty circle of radius 7 made of Brown Wool
  go 1 block up ↑
```



⚡ Create a Birthday Cake



Now we insert another layer in the cake, which means that after having inserted a brown layer we go one step up and add a white layer

```

/vm cake
go 10 block forward ↑
repeat 3 times
do
  create a empty circle of radius 7 made of Brown Wool
  go 1 block up ↑
  create a empty circle of radius 7 made of White Wool
  go 1 block up ↑

```



⚡ Create a Birthday Cake

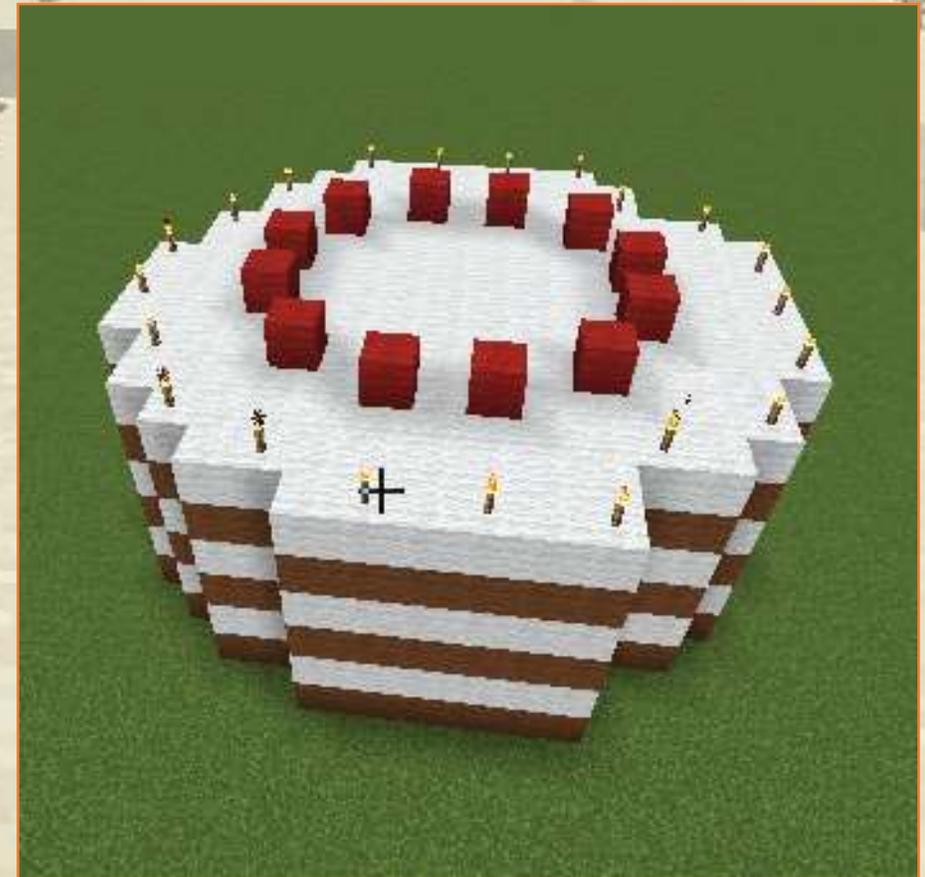


At the end of the program, outside the loop we add to circles on with lights the other with cherries. We alternate them with air to create spacing.

```

? /vm cake
go 10 block forward
repeat 3 times
do
  create a empty circle of radius 7 made of Brown Wool
  go 1 block up
  create a empty circle of radius 7 made of White Wool
  go 1 block up
  create a empty circle of radius 7 made of Torch Air
  create a empty circle of radius 4 made of Red Wool Air

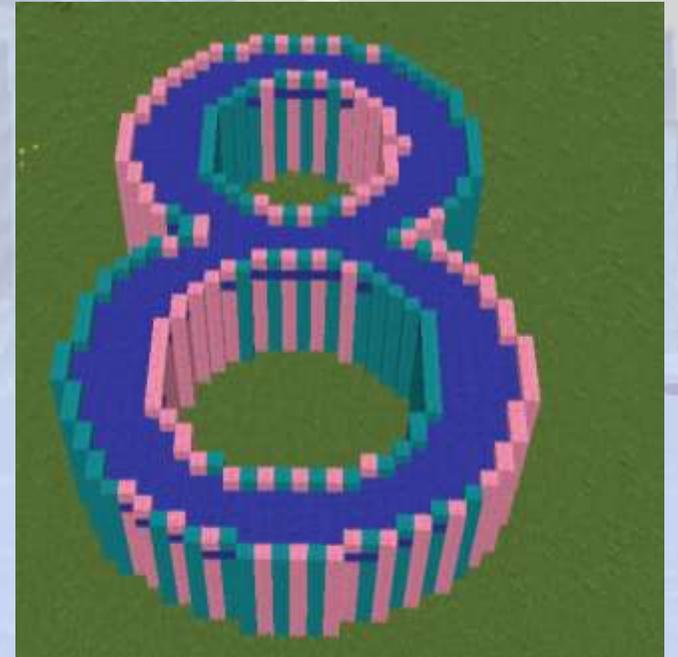
```



Combining Blocks



Create beautiful
structures by
combining blocks



Iteration with Simple Loops

Section Overview

We will explore interesting blocks and mobs, learning how to create and interact with them using our coding tools.

Objectives

Discover unique block types and their creative uses, along with interacting with mobs and custom structures.

Expected Outcomes



Blocks lists

We can mix objects by organizing them in a list

In the first program the robot will always use the same block but in the second program they are mixed

```
/vm list  
create a row of length 5 made of Magma Block
```



```
/vm list  
create a row of length 6 made of Block of Gold, Bookshelf, Bookshelf, Bookshelf, Obsidian, Quartz Stairs
```



Blocks lists

What happens if we tell robot to make the list longer?

```
/vm list  
create a row of length 6 made of Block of Gold Bookshelf Bookshelf Bookshelf Obsidian Quartz Stairs
```



```
/vm list  
create a row of length 7 made of Block of Gold Bookshelf Bookshelf Bookshelf Obsidian Quartz Stairs
```

Quiz



Blocks lists

The robot restarts the list from the beginning

```
/vm list  
create a row of length 6 made of Block of Gold Bookshelf Bookshelf Bookshelf Obsidian Quartz Stairs
```



```
/vm list  
create a row of length 7 made of Block of Gold Bookshelf Books
```





Blocks lists

We can avoid repeating many times the same Minecraft block.

This two programs have the same result but the second one is shorter

```
create a row of length 6 made of Block of Gold Bookshelf Bookshelf Bookshelf Obsidian Quartz Stairs
```



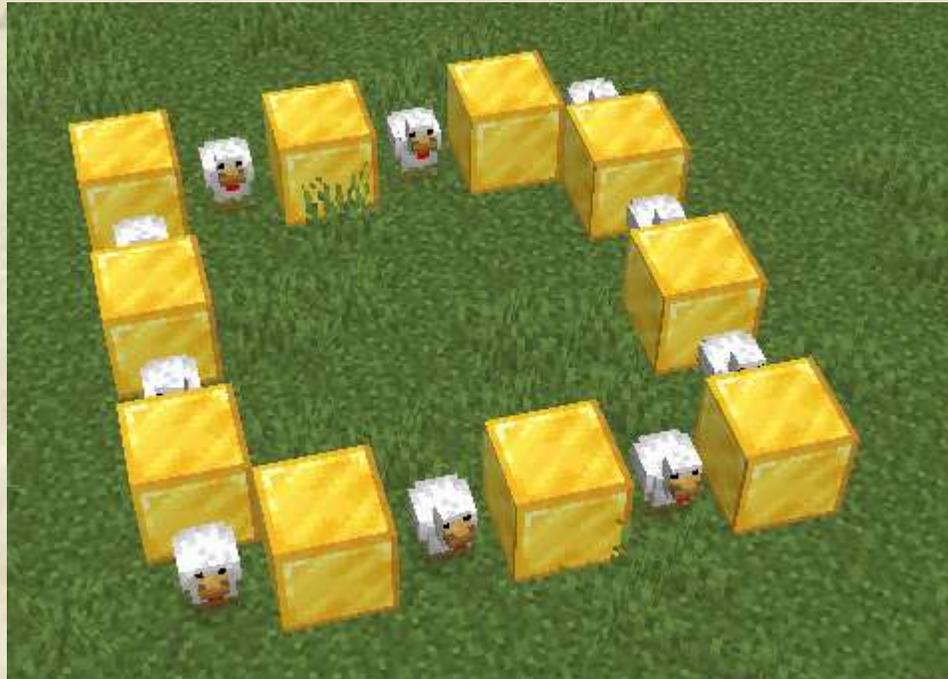
```
create a row of length 6 made of Block of Gold 3 of Bookshelf Obsidian Quartz Stairs
```



Blocks lists

Minecraft blocks can be combined with mobs

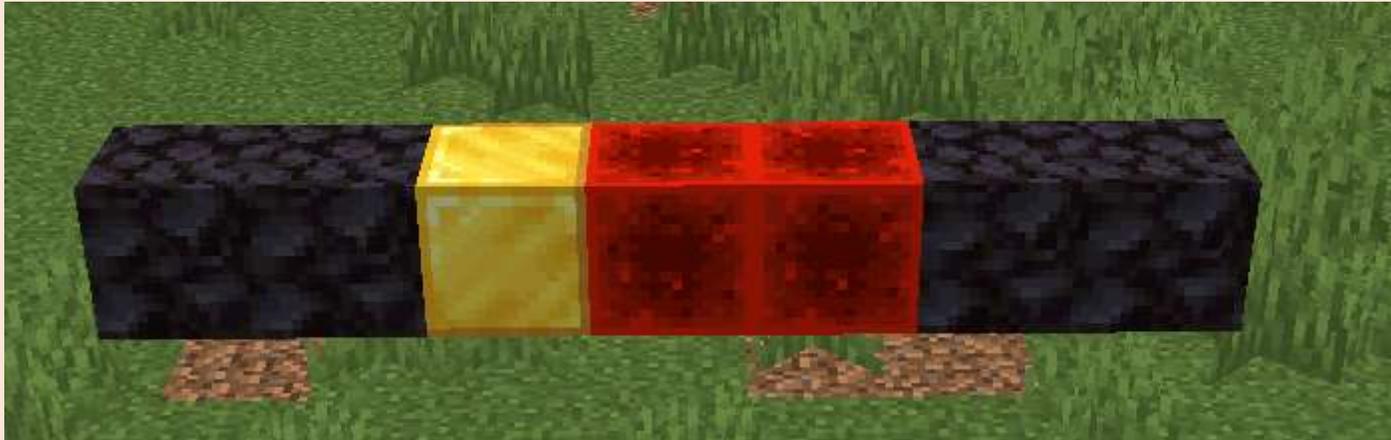
```
/vm square  
create a empty square of width 6 made of Block of Gold Chicken
```





What blocks are created by these programs?

A, B or C?



A

```
/vm test  
create a row -- of length 4 made of 2 of Blackstone Block of Gold 2 of Block of Redstone
```

B

```
/vm test  
create a row -- of length 7 made of 2 of Blackstone Block of Gold 2 of Block of Redstone
```

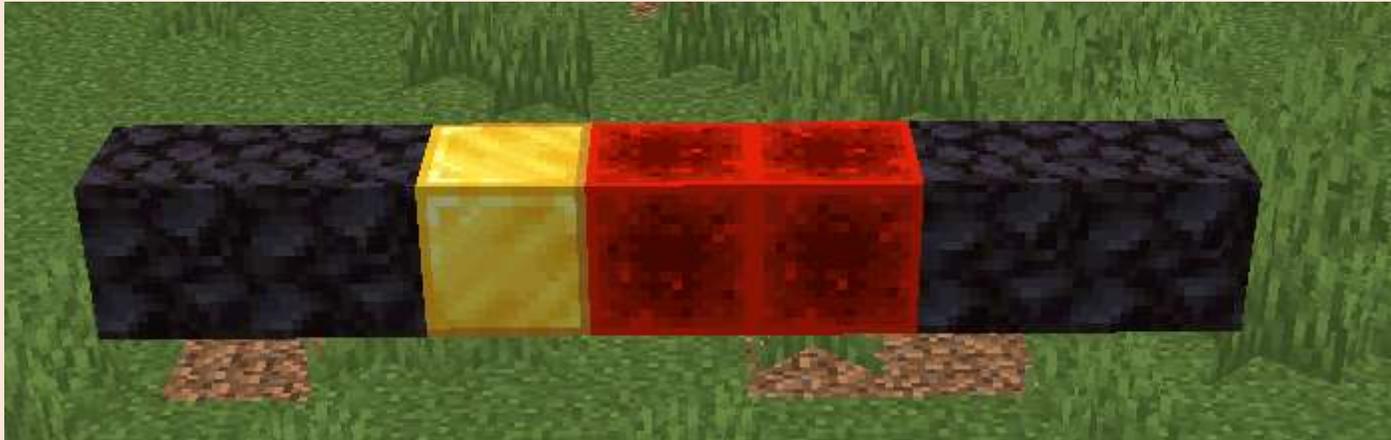
C

```
/vm test  
create a row -- of length 7 made of 2 of Blackstone Block of Gold Block of Redstone
```

Quiz

What blocks are created by these programs?

A, B or C?



A

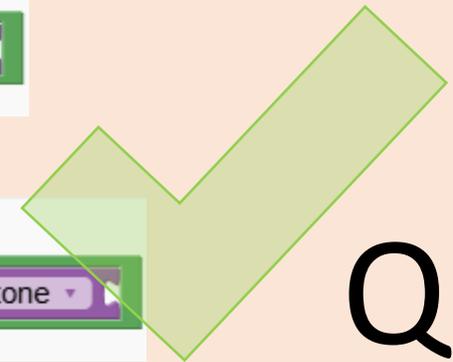
```
code blocks: /vm test, create a row -- of length 4 made of 2 of Blackstone, Block of Gold, 2 of Block of Redstone
```

B

```
code blocks: /vm test, create a row -- of length 7 made of 2 of Blackstone, Block of Gold, Block of Redstone
```

C

```
code blocks: /vm test, create a row -- of length 7 made of 2 of Blackstone, Block of Gold, 2 of Block of Redstone
```



Quiz

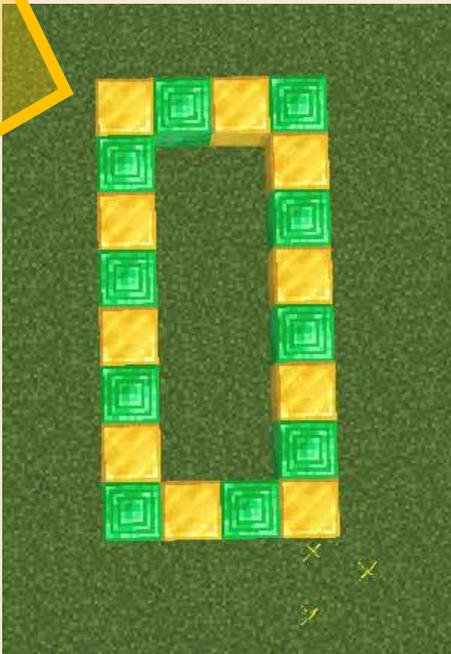


What does this program create?

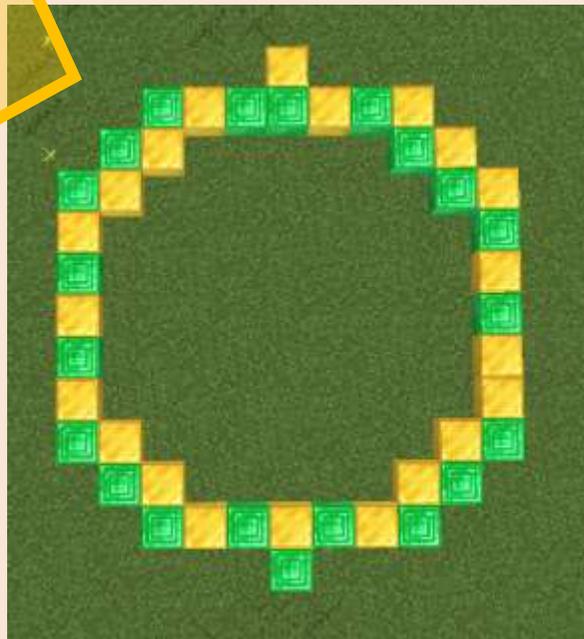
A, B or C?

```
create a empty circle of radius 6 made of Block of Gold Block of Emerald
```

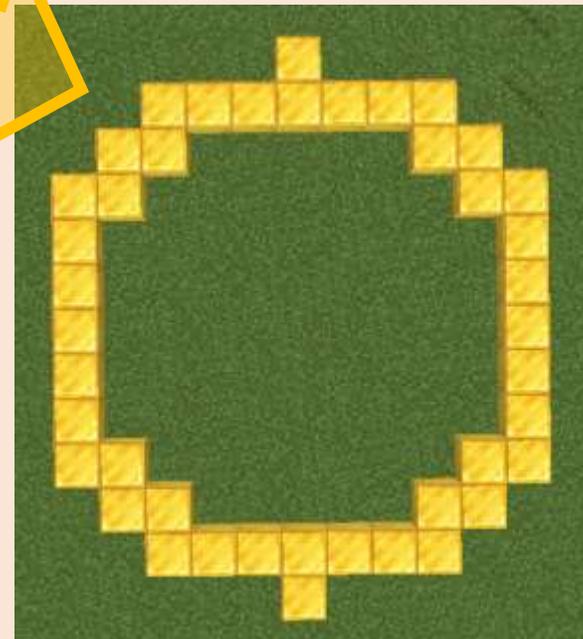
A



B



C



Quiz

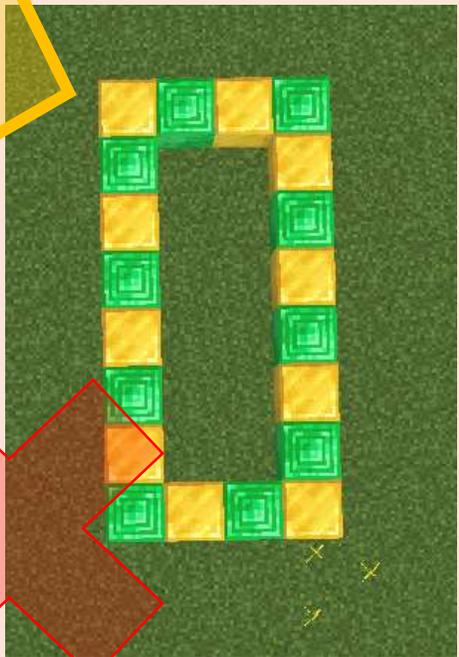


What does this program create?

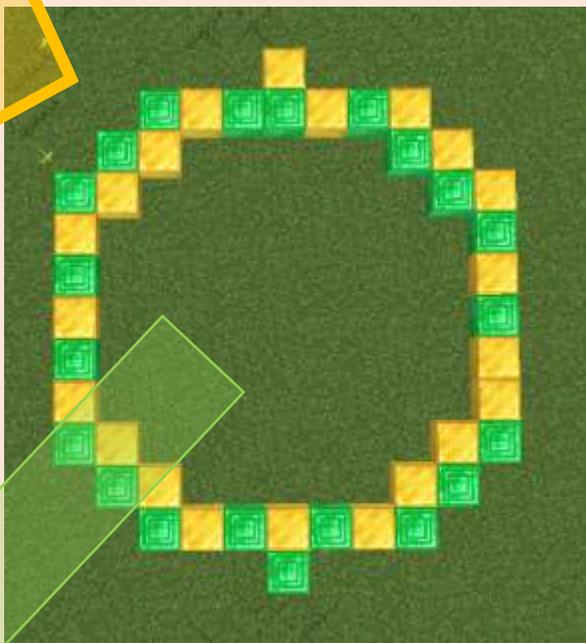
A, B or C?

```
create a empty circle of radius 6 made of Block of Gold Block of Emerald
```

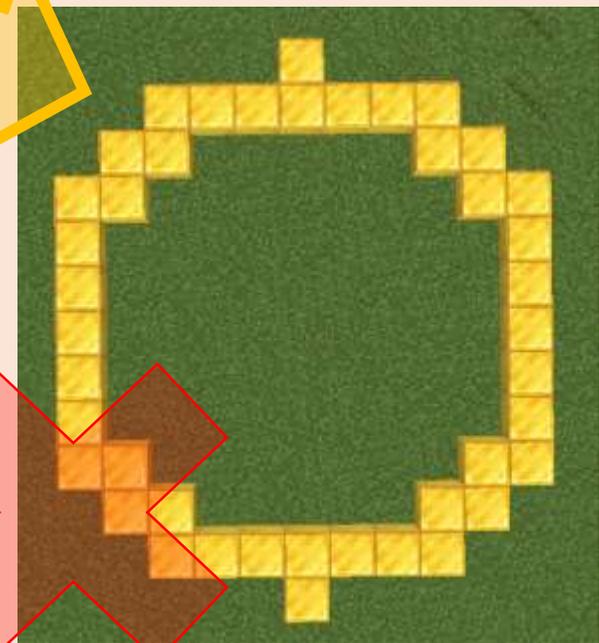
A



B



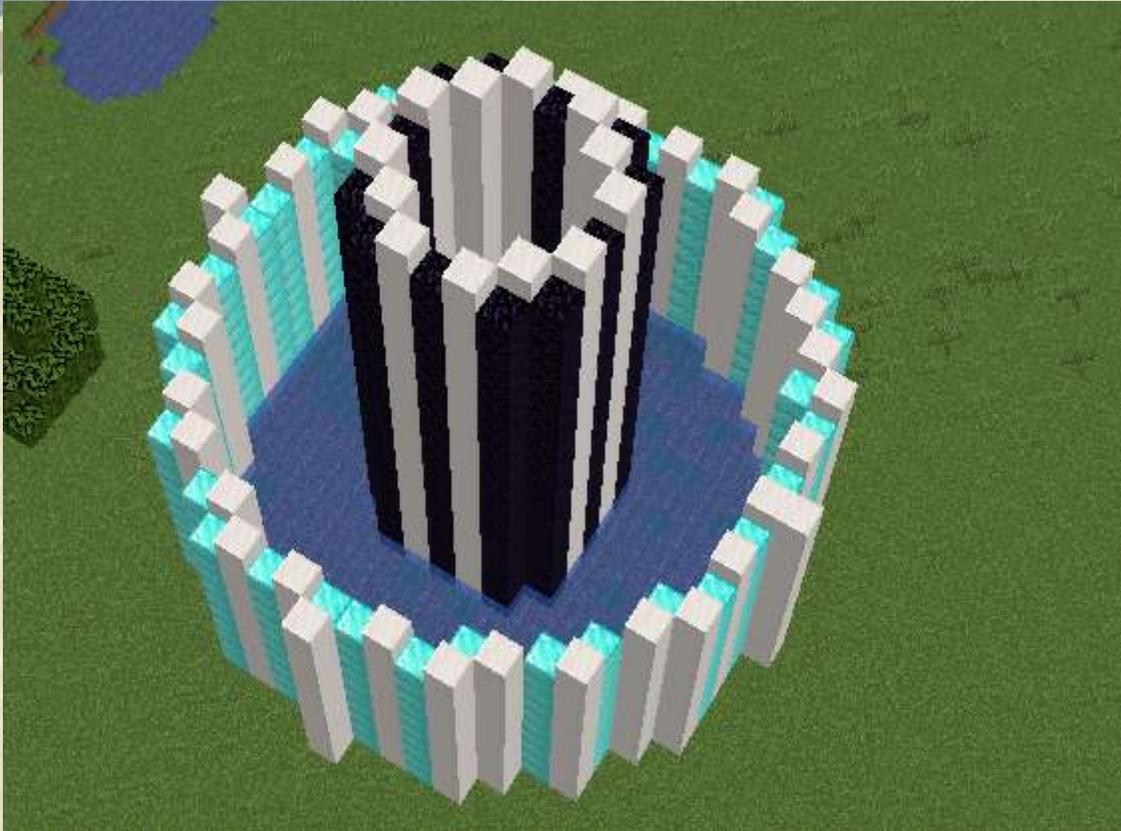
C



Quiz

⚡ Make your own castle

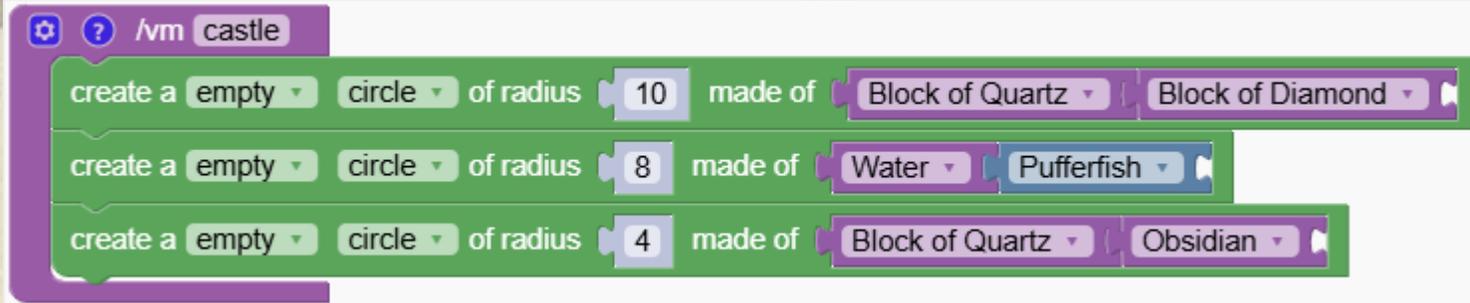
We are building a own castle by combining blocks and Mobs



⚡ Make your own castle

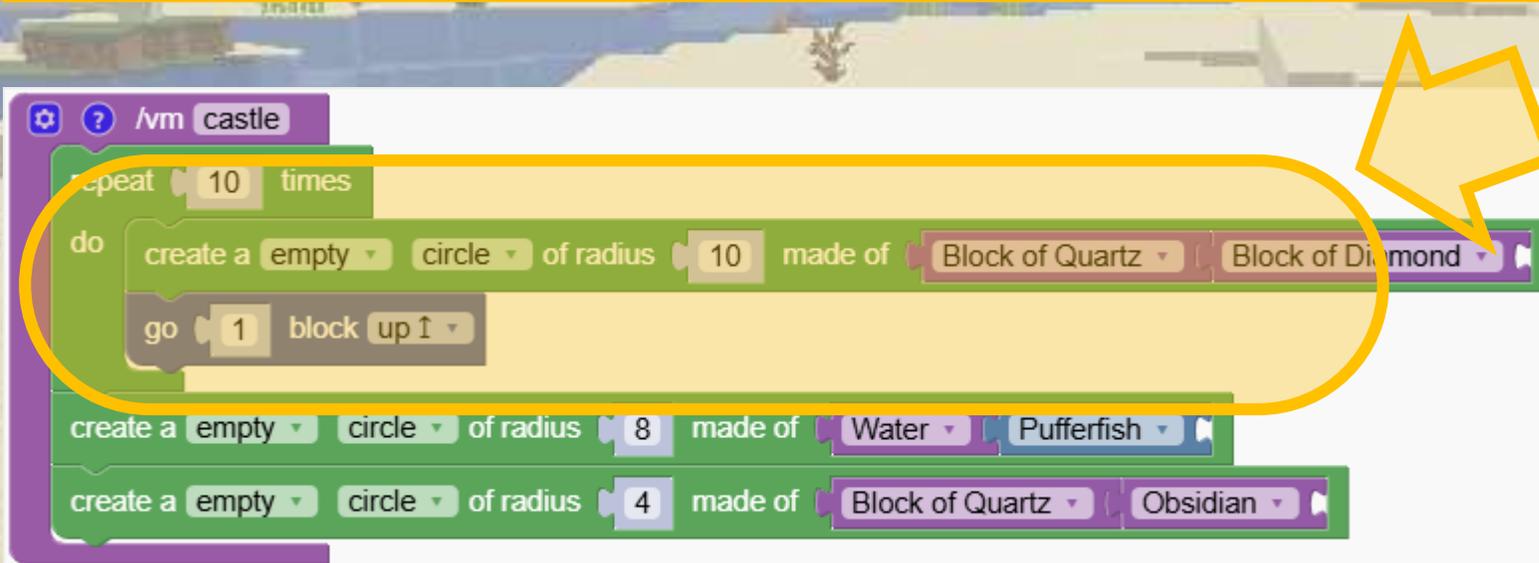
First we create the base of the castle.

We added some pufferfish. Don't go too close to them!

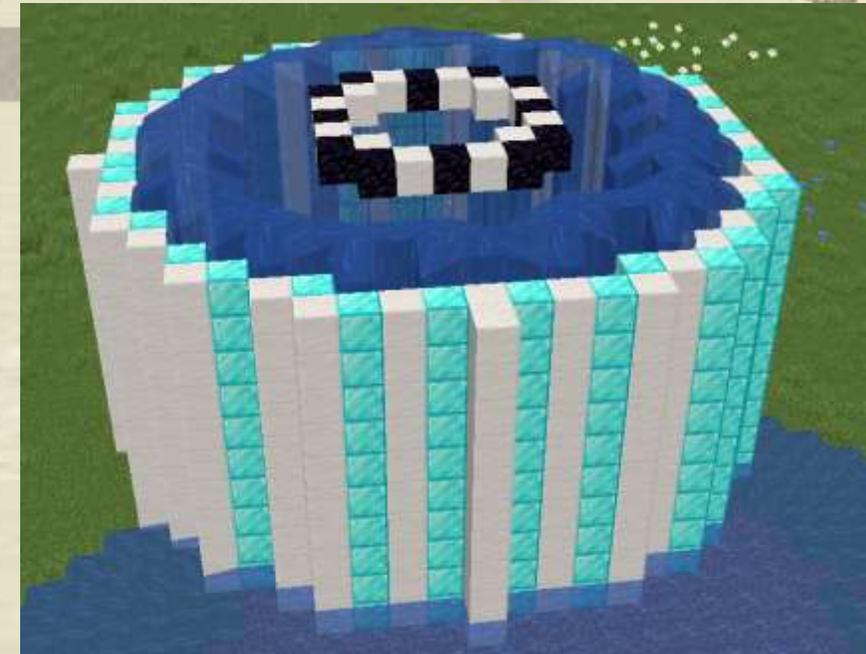


⚡ Make your own castle

We add a loop to make the outside wall grow. But something is wrong



```
repeat 10 times
do
  create a empty circle of radius 10 made of Block of Quartz Block of Diamond
  go 1 block up
create a empty circle of radius 8 made of Water Pufferfish
create a empty circle of radius 4 made of Block of Quartz Obsidian
```



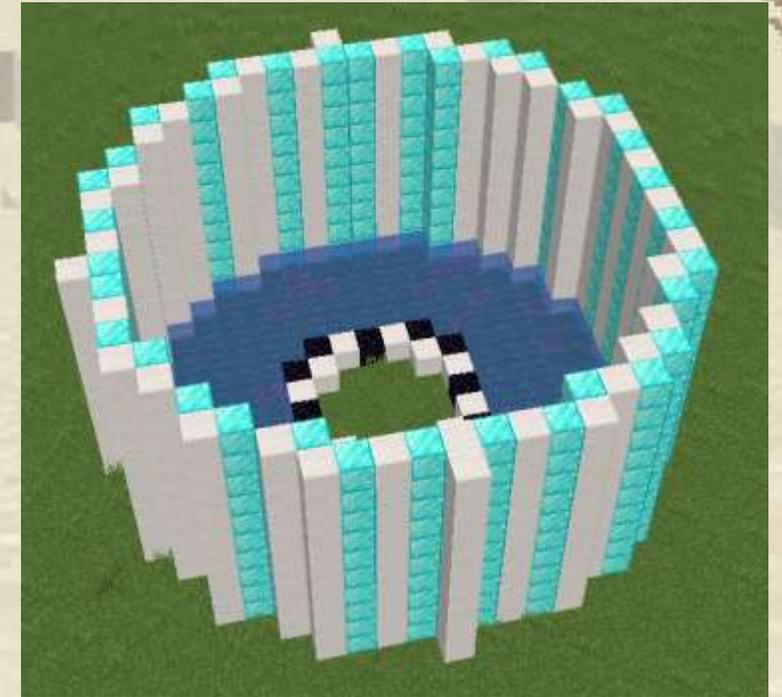
The robots moves all the way to the top of the outside wall and then continues from there to make the inner wall.

We have to tell robot to go back to the start position before doing the water and the inner wall

⚡ Make your own castle

We will explore interesting blocks and mobs, learning how to create and interact with them using our coding tools.

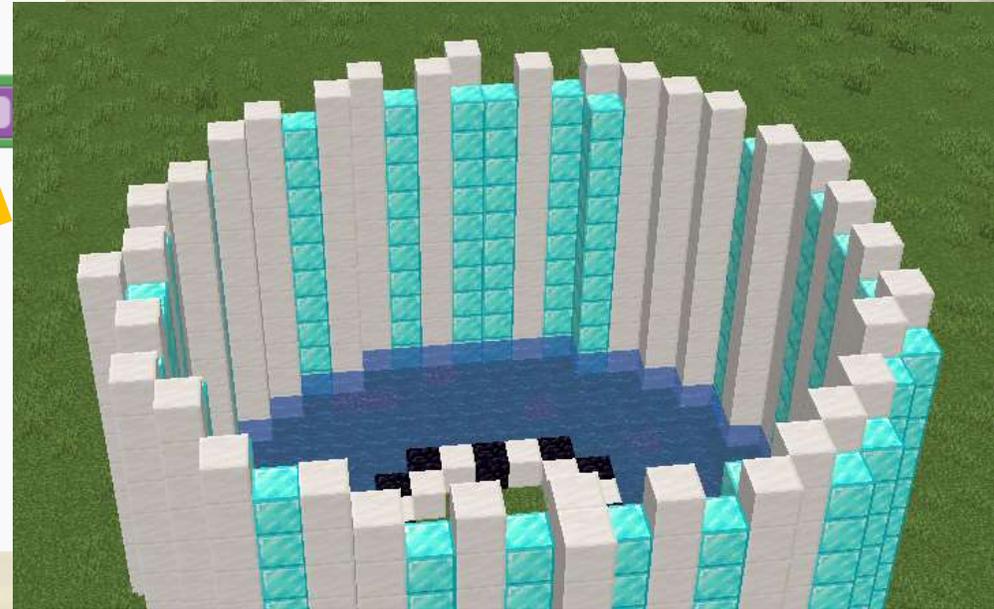
```
vm castle
repeat 10 times
do
  create a empty circle of radius 10 made of Block of Quartz Block of Diamond
  go 1 block up
go to the start
create a empty circle of radius 8 made of Water Pufferfish
create a empty circle of radius 4 made of Block of Quartz Obsidian
```



⚡ Make your own castle

We would like to have some crown like edge on the walls
So we add a level made of quartz and air

```
vm castle
repeat 10 times
do
  create a empty circle of radius 10 made of Block of Quartz Block of Diamond
  go 1 block up
  create a empty circle of radius 10 made of Block of Quartz Air
  go to the start
  create a empty circle of radius 8 made of Water Pufferfish
  create a empty circle of radius 4 made of Block of Quartz Obsidian
```



⚡ Make your own castle

To finish we repeat the inner circle to become a tower in the same way as the outer tower

```

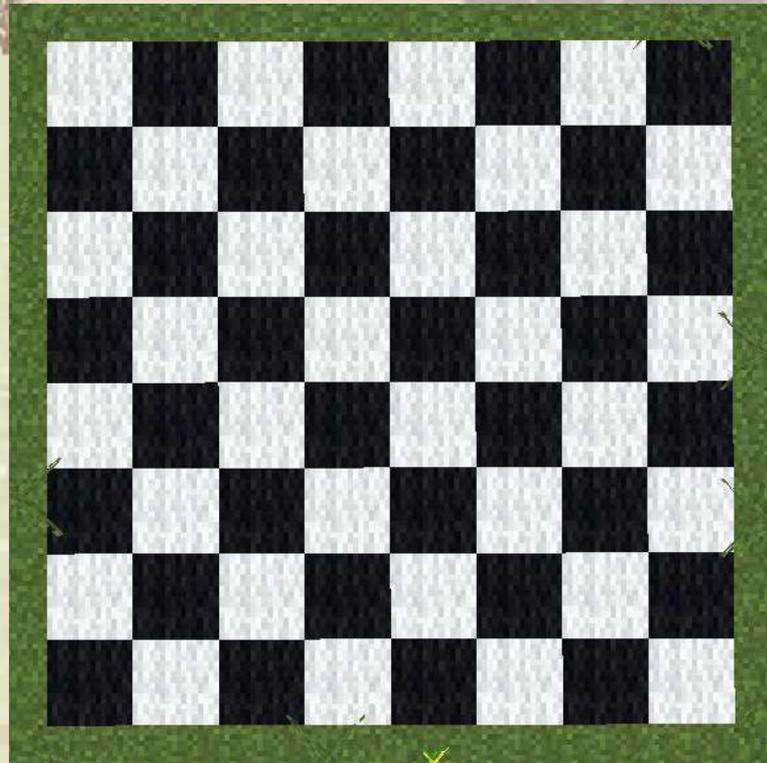
/mv castle
repeat 10 times
do
  create a empty circle of radius 10 made of Block of Quartz Block of Diamond
  go 1 block up ↑
create a empty circle of radius 10 made of Block of Quartz Air
go to the start
create a empty circle of radius 8 made of Water Pufferfish
repeat 20 times
do
  create a empty circle of radius 4 made of Block of Quartz Obsidian
  go 1 block up ↑
create a empty circle of radius 4 made of Block of Quartz Air

```



Chess Board

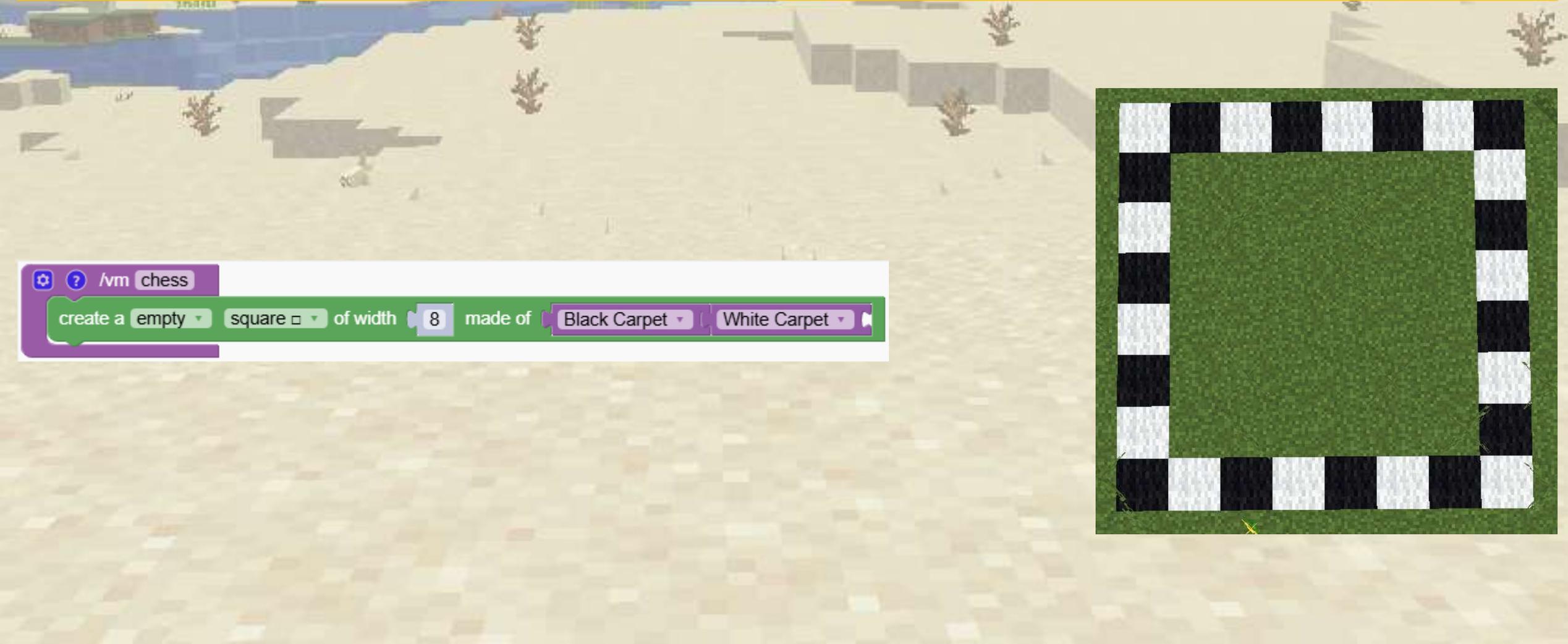
How do we create a chessboard?



⚡ Chess Board

We are building a chess board.

First we create the outside blocks by alternating black and white



```
lvn chess  
create a empty square of width 8 made of Black Carpet White Carpet
```



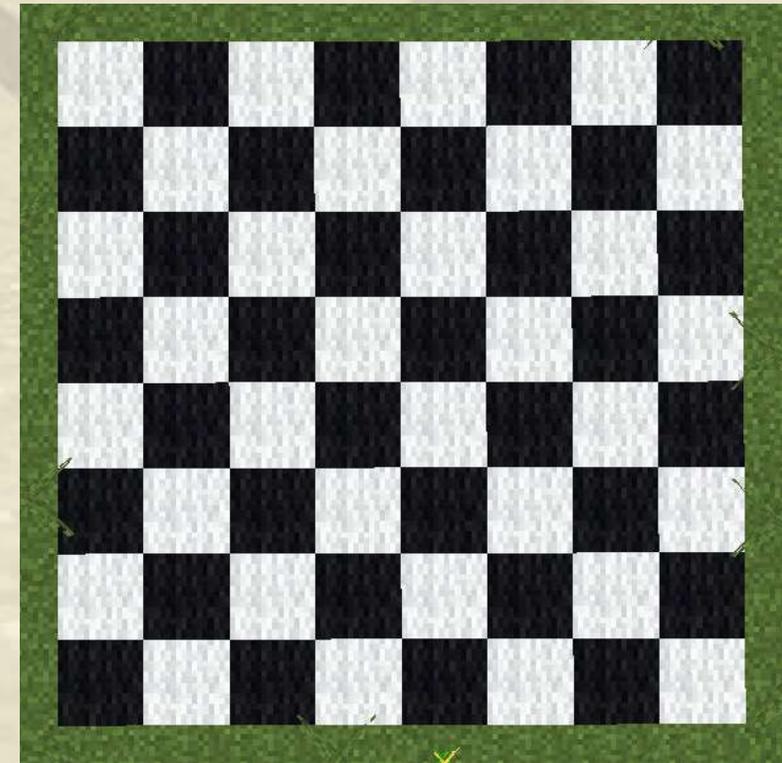
⚡ Chess Board

We add the smaller squares inside.

```

? /vm chess
create a empty square of width 8 made of Black Carpet White Carpet
create a empty square of width 6 made of Black Carpet White Carpet
create a empty square of width 4 made of Black Carpet White Carpet
create a empty square of width 2 made of Black Carpet White Carpet

```



⚡ Use letters to create fun structures

We are able to write text with blocks:
You can customize it's appearance.

```
/vm text  
write " abc " using font Monospace plain of size 60 points, full , made of Pink Wool Cyan Wool
```



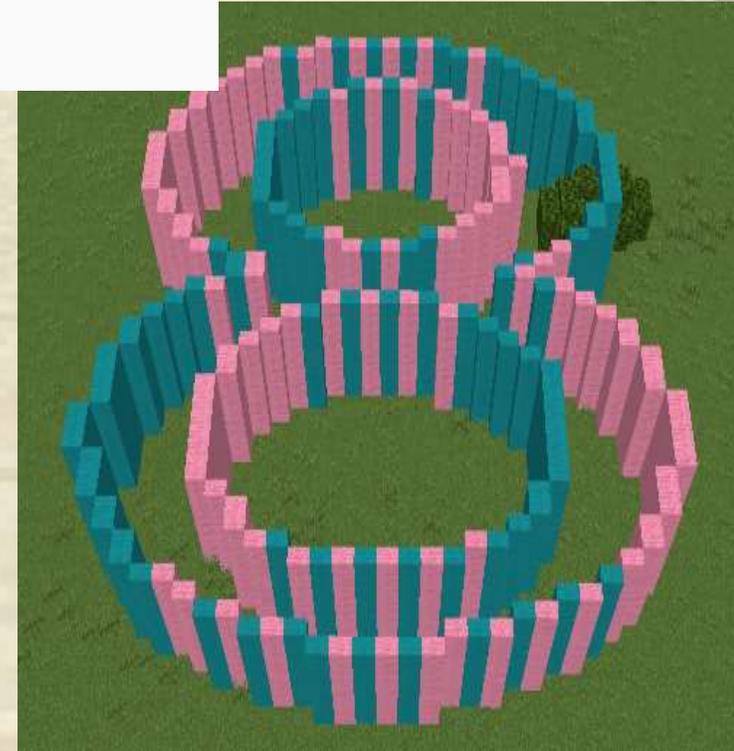
```
/vm text  
write " abc " using font Monospace plain of size 60 points, empty , made of Pink Wool Cyan Wool
```



⚡ Use letters to create fun structures

We are making a cattle in the shape of a number 8

```
repeat 10 times
do
  write " 8 " using font Monospace plain of size 60 points, empty, made of Pink Wool Cyan Wool
  go 1 blocks up ↑
```



⚡ Use letters to create fun structures

We are making a castle in the shape of a number

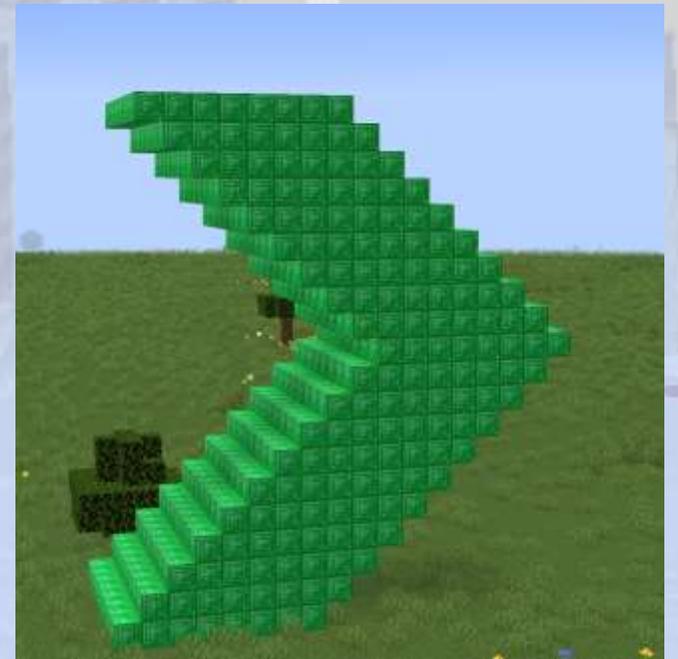
```
repeat 10 times
do
  write "8" using font Monospace plain of size 60 points, empty, made of Pink Wool Cyan Wool
  go 1 blocks up ↑
go 2 blocks down ↓
write "8" using font Monospace plain of size 60 points, full, made of Blue Wool
```



Moving in the world



Learn how to control
the robot position



3D orientation

Section Overview

We will understand turtle movements and practice 3D positioning by creating fun designs.

Objectives

Master positioning blocks in 3D space using movement blocks and structured learning exercises.

Expected Outcomes



The movement blocks

The robot can be moved in the world

```
 /vm move1  
 create a block = made of Blackstone  
 go 1 blocks up ↑  
 create a block = made of Block of Gold
```



```
 /vm move2  
 create a block = made of Blackstone  
 go 1 blocks left ←  
 create a block = made of Block of Gold
```



```
 /vm move3  
 create a block = made of Blackstone  
 go 1 blocks forward ↑  
 create a block = made of Block of Gold
```



⚡ Let's create a smiley face

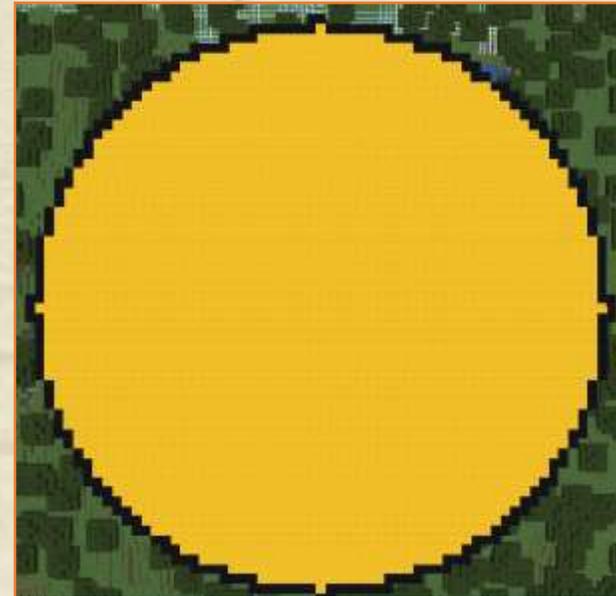
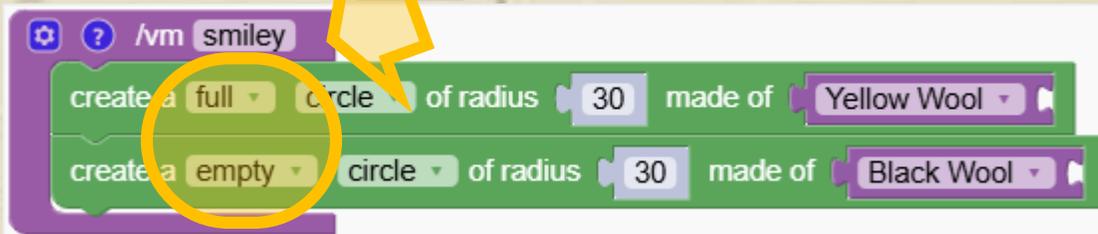
Learn block placement by building a simple and fun smiley face.
The students will have fun by customizing it



⚡ Let's create a smiley face

Lets create two circles of radius 30 blocks.

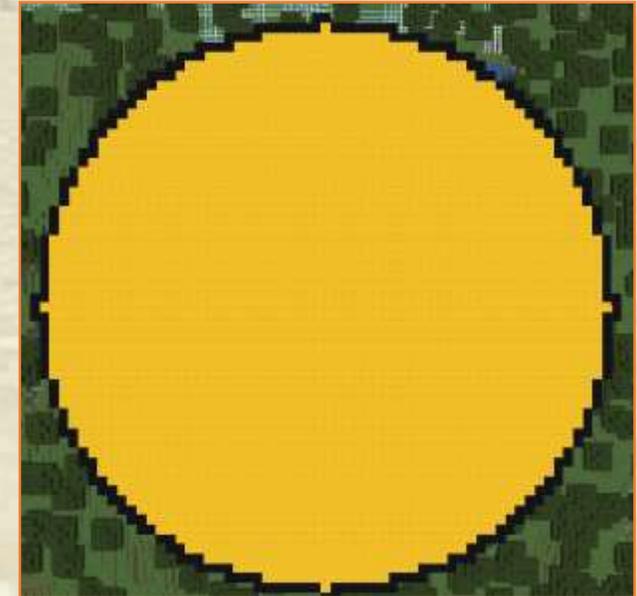
The first one is full and the second one is empty because it is just a black border



⚡ Let's create a smiley face

Every time we create the circles we have to fly high up in order to see the whole smiley.
To be more efficient, we want to fly high above ground and look at the smiley faces from there.
Therefore we use this block to say that the block should be on top of the first solid block found

```
vm smiley
create a full circle of radius 30 made of on the ground Yellow Wool
create a empty circle of radius 30 made of on the ground Black Wool
```



⚡ Let's create a smiley face

Our robot is located at the center of the face.

Now we give the order to move 10 steps forward and 15 step right before making the circles for the eye

```

? /vm smiley
create a full circle of radius 30 made of on the ground Yellow Wool
create a empty circle of radius 30 made of on the ground Black Wool
go 10 block forward ↑
go 15 block right →
create a full circle of radius 5 made of on the ground Light Blue Wool
create a empty circle of radius 5 made of on the ground Black Wool

```



⚡ Let's create a smiley face

Now we give the order to move 25 steps left and make 2 circles of with 6 blocks

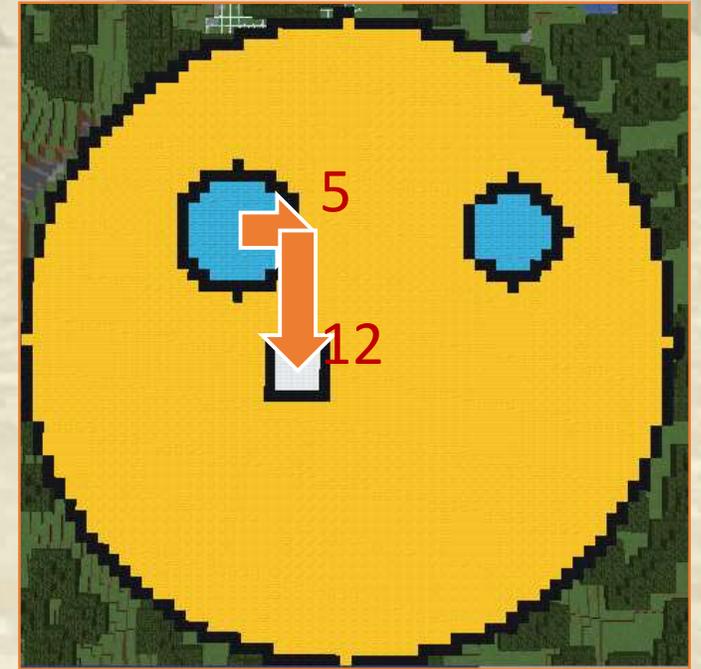
```
vm smiley
create a full circle of radius 30 made of on the ground Yellow Wool
create a empty circle of radius 30 made of on the ground Black Wool
go 10 block forward
go 15 block right
create a full circle of radius 5 made of on the ground Light Blue Wool
create a empty circle of radius 5 made of on the ground Black Wool
go 25 block left
create a full circle of radius 6 made of on the ground Light Blue Wool
create a empty circle of radius 6 made of on the ground Black Wool
```



⚡ Let's create a smiley face

Now we give the order to move 25 steps left and to make 2 circles of with 6 blocks

```
create a full circle of radius 30 made of on the ground Yellow Wool
create a empty circle of radius 30 made of on the ground Black Wool
go 10 block forward ↑
go 15 block right →
create a full circle of radius 5 made of on the ground Light Blue Wool
create a empty circle of radius 5 made of on the ground Black Wool
go 25 block left ←
create a full circle of radius 6 made of on the ground Light Blue Wool
create a empty circle of radius 6 made of on the ground Black Wool
go 5 block right →
go 12 block backwards ↓
create a full square of width 6 made of on the ground White wool
create a empty square of width 6 made of on the ground Black Wool
```



⚡ Let's create a smiley face

Now we give the order to move 12 blocks backwards

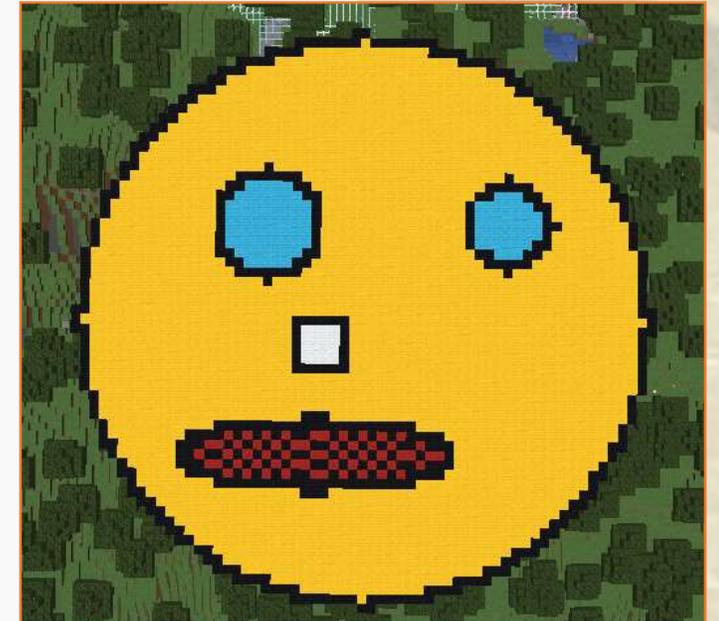
```
go 15 block right →
create a full circle of radius 5 made of on the ground Light Blue Wool
create a empty circle of radius 5 made of on the ground Black Wool
go 25 block left ←
create a full circle of radius 6 made of on the ground Light Blue Wool
create a empty circle of radius 6 made of on the ground Black Wool
go 5 block right →
go 12 block backwards ↓
create a full square of width 6 made of on the ground White Wool
create a empty square of width 6 made of on the ground Black Wool
go 12 block backwards ↓
create a full ellipse with radiusX 15 and radiusY 4 made of on the ground Red Wool on the ground Black Wool
create a empty ellipse with radiusX 15 and radiusY 4 made of on the ground Black Wool
```



⚡ Let's create a smiley face

Now is finished

```
create a full circle of radius 5 made of on the ground Light Blue Wool
create a empty circle of radius 5 made of on the ground Black Wool
go 25 block left ←
create a full circle of radius 6 made of on the ground Light Blue Wool
create a empty circle of radius 6 made of on the ground Black Wool
go 5 block right →
go 12 block backwards ↓
create a full square of width 6 made of on the ground White Wool
create a empty square of width 6 made of on the ground Black Wool
go 12 block backwards ↓
create a full ellipse with radiusX 15 and radiusY 4 made of on the ground Red Wool on the ground Black Wool
create a empty ellipse with radiusX 15 and radiusY 4 made of on the ground Black Wool
```



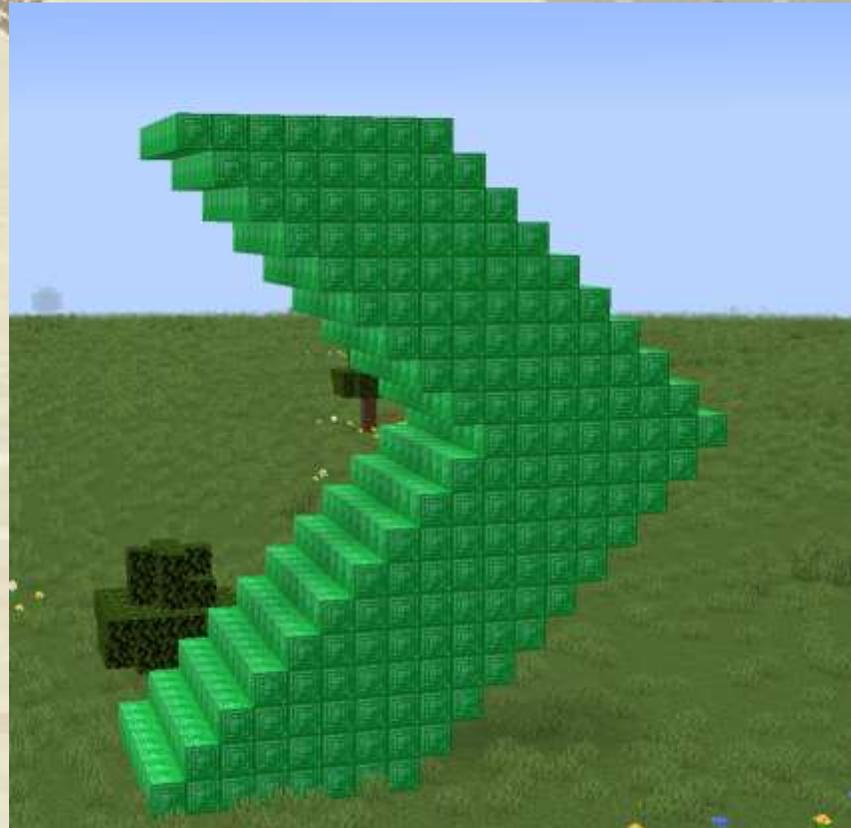
Independent time: Make your own smiley

Have fun customizing the program and making your own smiley faces.



⚡ We make a tower in the shape of an arrow

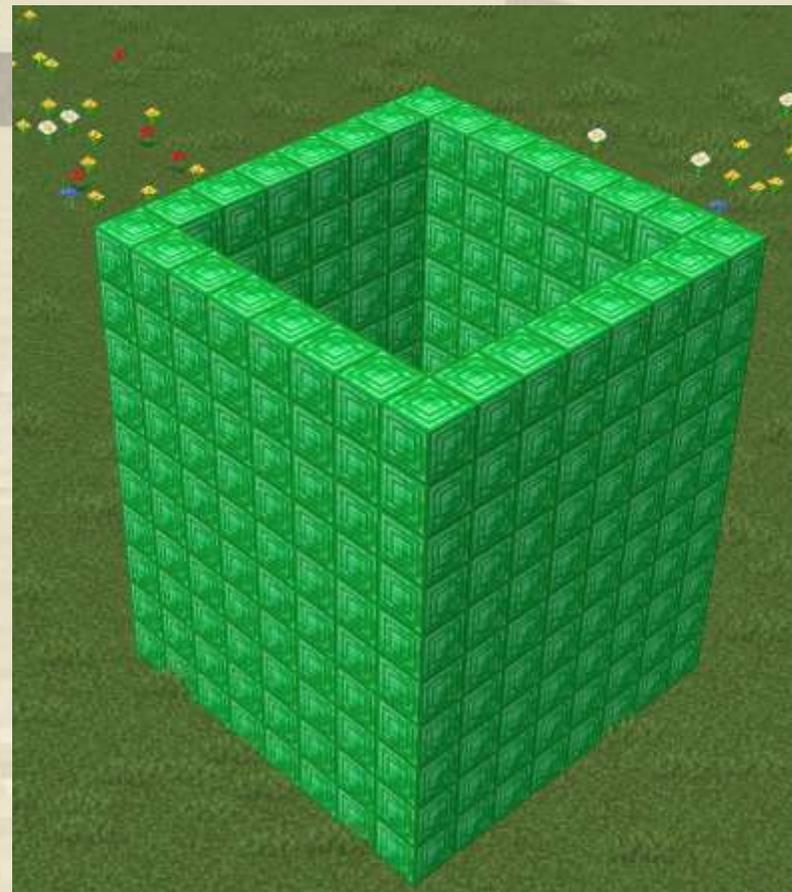
Combine movements to construct a tower shaped like an arrow.



⚡ We make a tower in the shape of an arrow

Let's start by creating a simple emerald tower

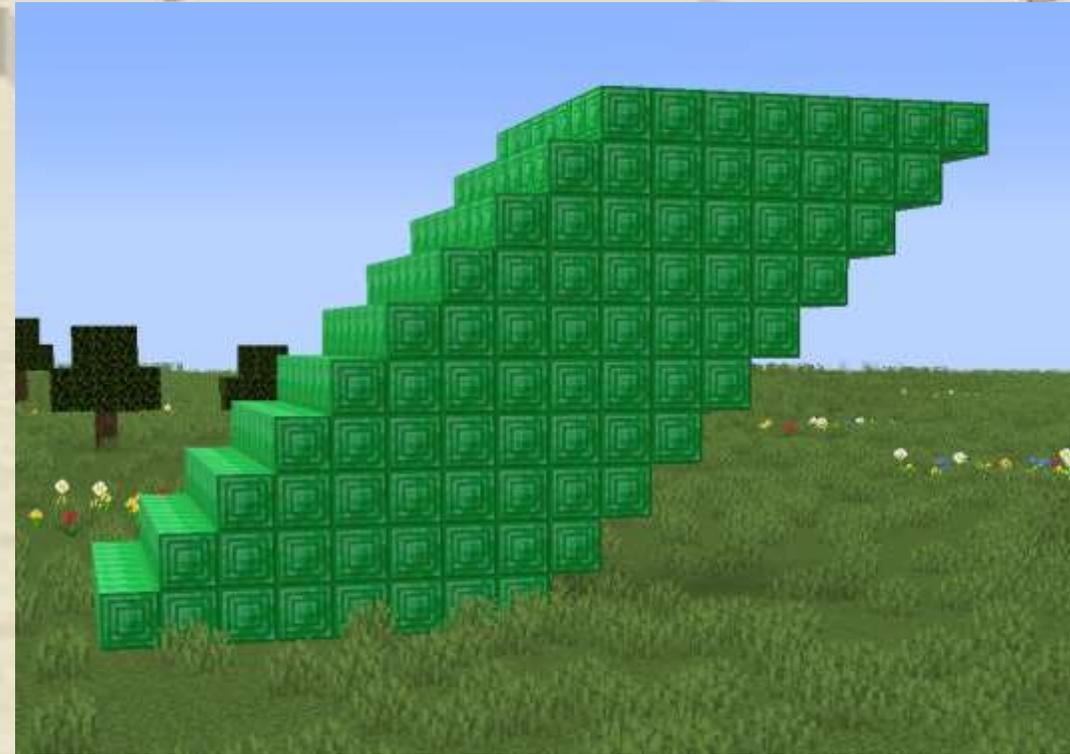
```
go 10 block forward ↑
repeat 10 times
do
  create a empty square of width 8 made of Block of Emerald
  go 1 block up ↑
```



⚡ We make a tower in the shape of an arrow

We add a step forward to make the tower grow forward

```
vm arrow
go 10 block forward ↑
repeat 10 times
do
  create a empty square of width 8 made of Block of Emerald
  go 1 block up ↑
  go 1 block forward ↑
```



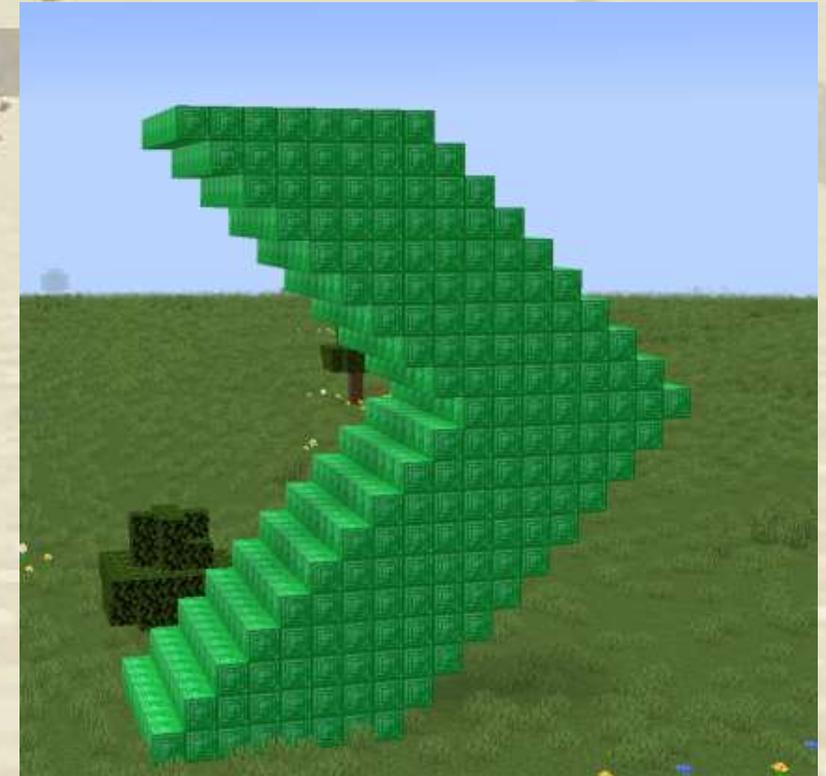
⚡ We make a tower in the shape of an arrow

Now we want to add another tower that goes in the other direction.
We duplicate the repeating block and we change the direction of the tower

```

/vm arrow
go 10 block forward ↑
repeat 10 times
do
  create a empty square of width 8 made of Block of Emerald
  go 1 block up ↑
  go 1 block forward ↑
repeat 10 times
do
  create a empty square of width 8 made of Block of Emerald
  go 1 block up ↑
  go 1 block backwards ↓

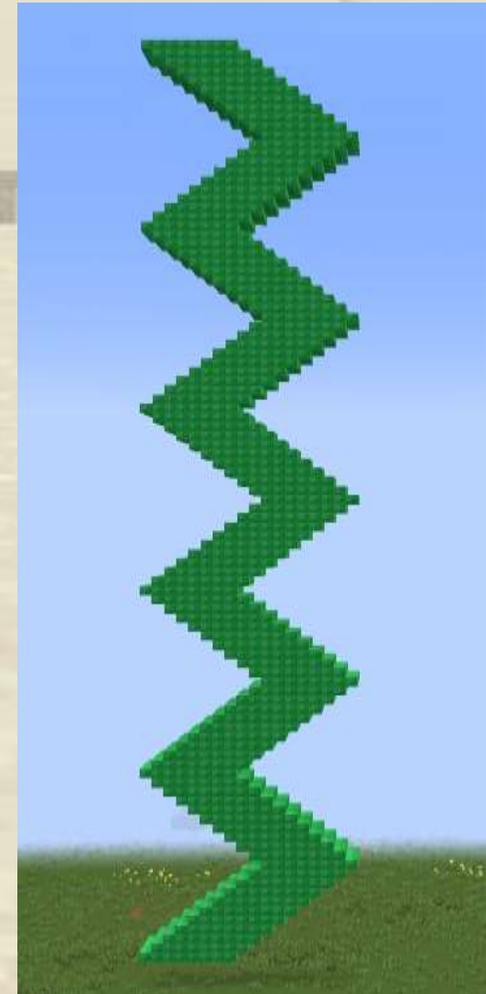
```



⚡ We make a tower in the shape of an arrow

Let's do an amazing tower. Just add a repeating block around the whole program and our tower becomes amazing!

```
go 10 block forward ↑
repeat 10 times
do
repeat 10 times
do
create a empty square of width 8 made of Block of Emerald
go 1 block up ↑
go 1 block forward ↑
repeat 10 times
do
create a empty square of width 8 made of Block of Emerald
go 1 block up ↑
go 1 block backwards ↓
```



How do we make a chicken bomb?

A chicken bomb is made with 30 chickens all spawned at the same position in the air. When the chickens land on the ground they spread similar to an explosion



Quiz

How do we make a chicken bomb?

Here is the solution.

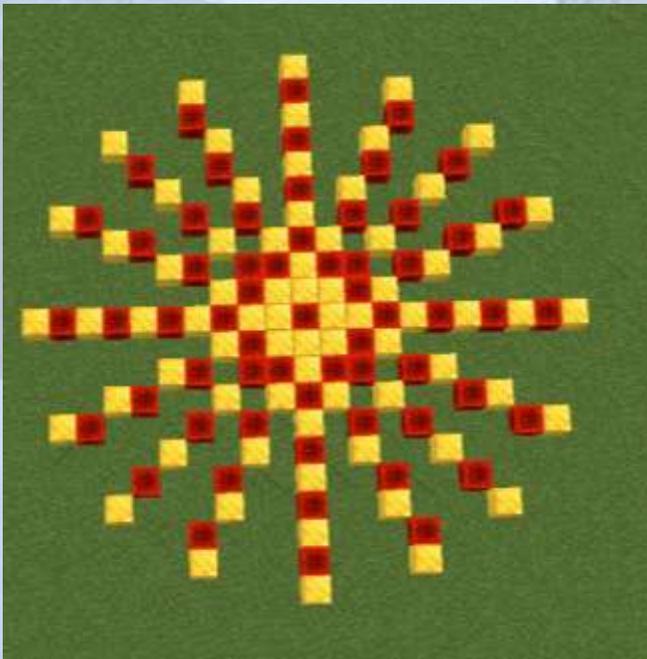
What happens if we use another mob instead of chickens?

```
script /vm bomb
  go 10 block up 1
  repeat 30 times
    do create a block made of Chicken
```



Quiz

Horizontal Rotation



Amazing structures
created with simple
rotations



Horizontal rotation

 Section Overview

 Objectives

Learn how to rotate objects horizontally and create interesting shapes like spirals and patterns.

 Expected Outcomes

Angles

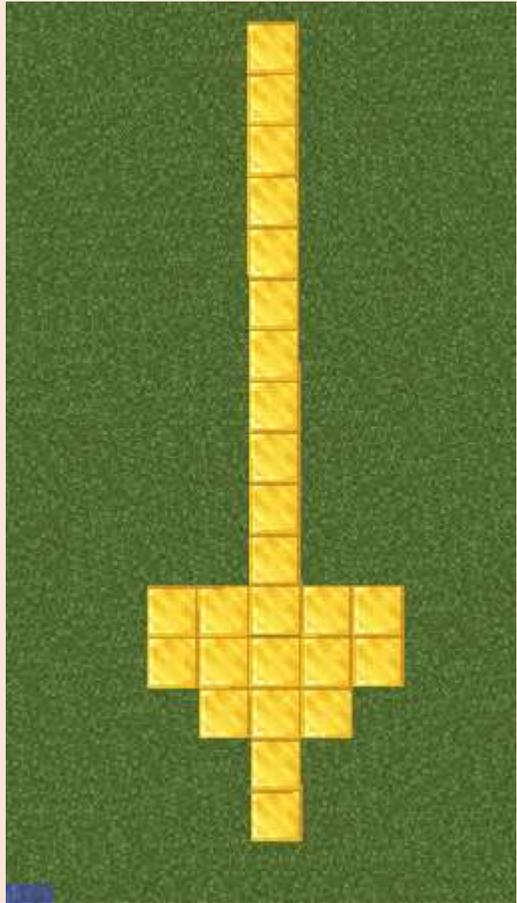
Review the concept of angles



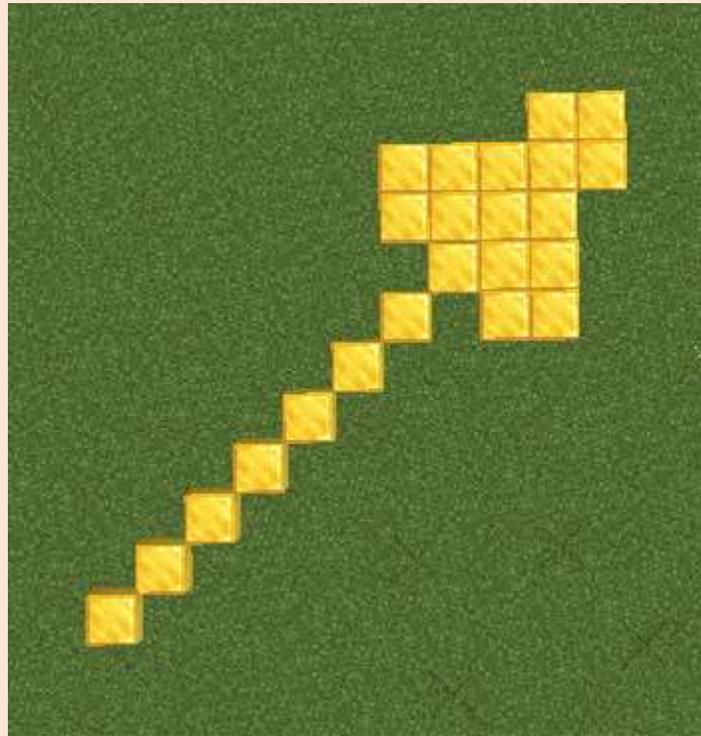
turn right by degrees

Which angle is this?

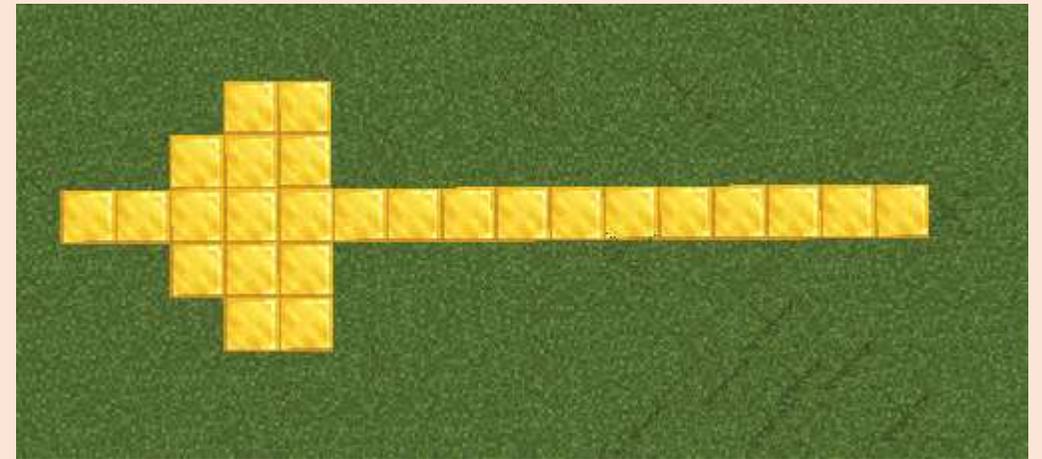
Test your knowledge of angles with this activity.



?



?

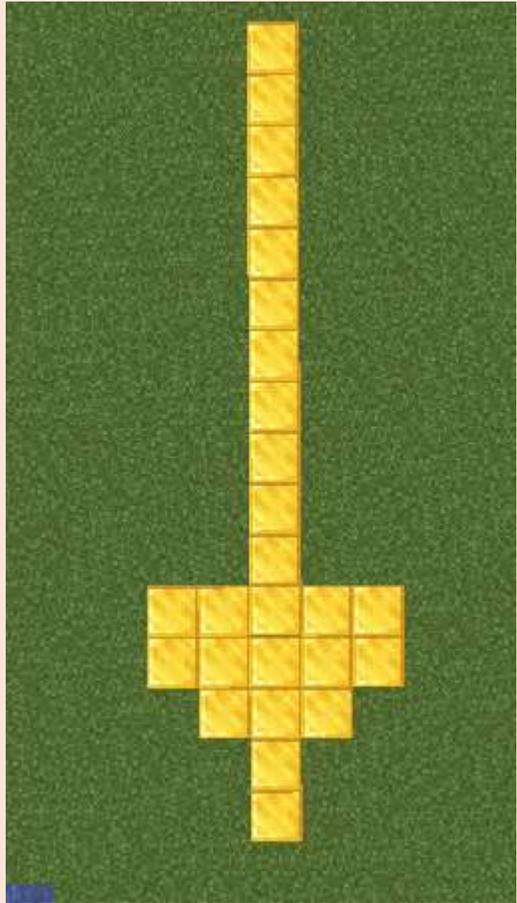


?

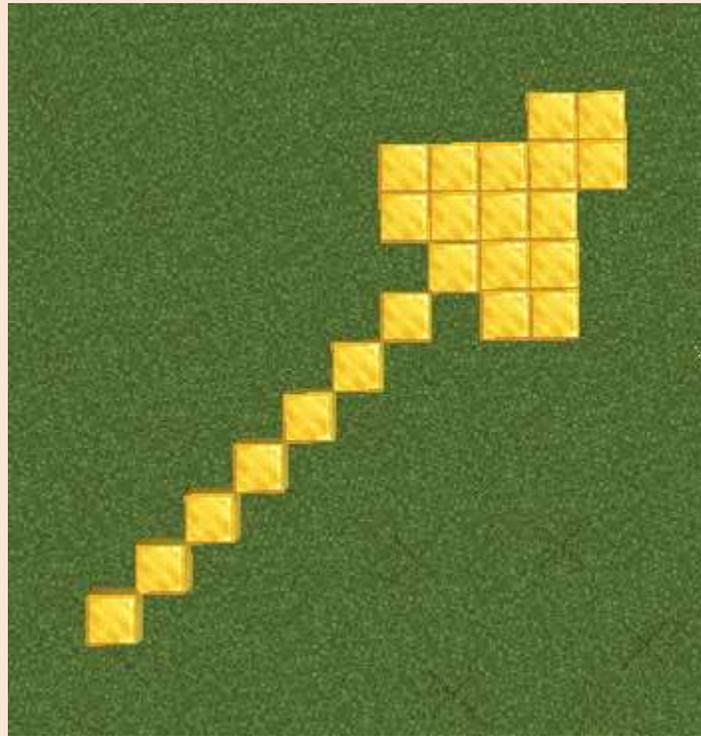
Quiz

Which angle is this?

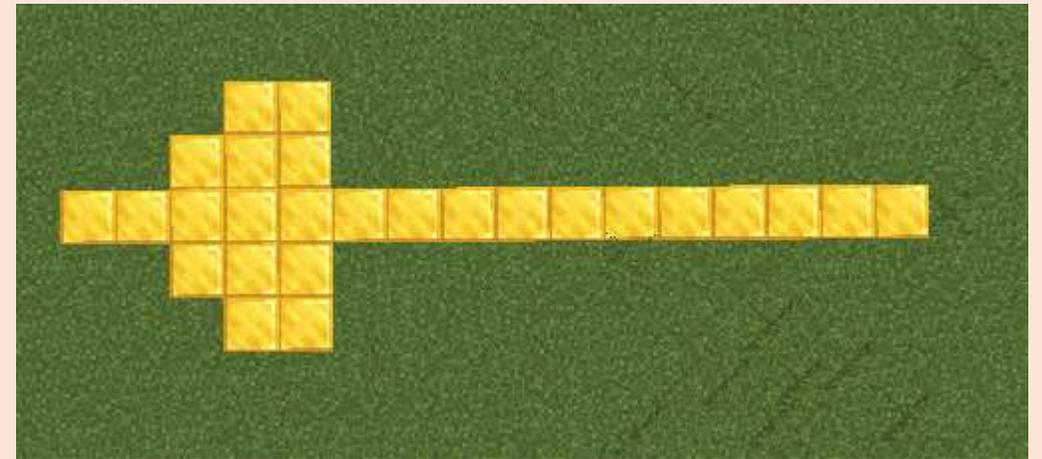
Solution:



180



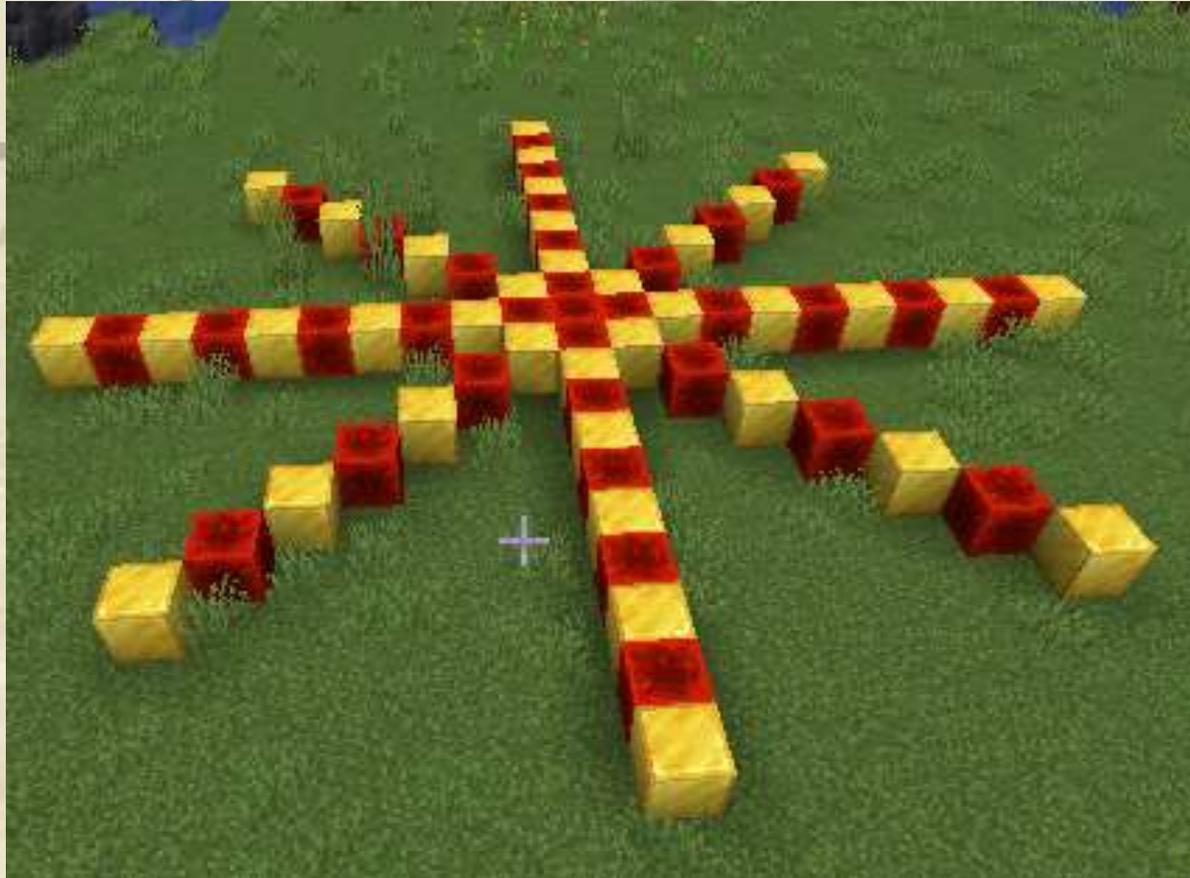
45



270 Quiz

⚡ Create a sun by rotating a row of blocks

Use simple rotation to create a fun sun drawing.



⚡ Create a sun by rotating a row of blocks

We start with one line of blocks.

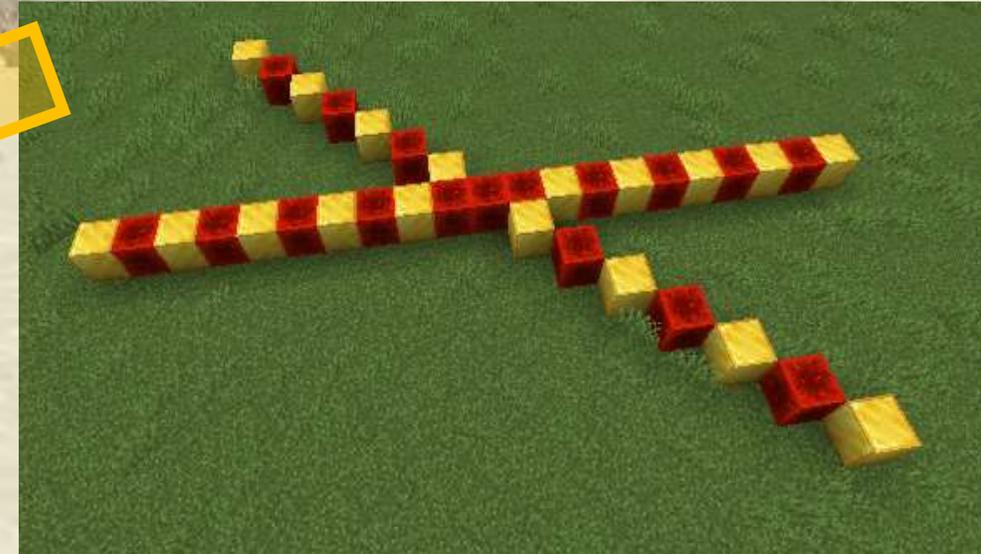
```
vm sun  
create a row of length 21 made of Block of Gold Block of Redstone
```



⚡ Create a sun by rotating a row of blocks

We add a second line after having done a 45 degrees rotation

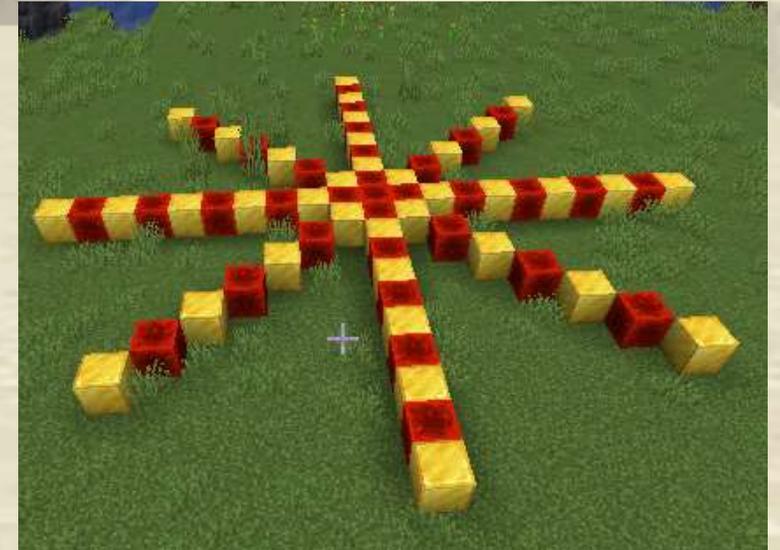
```
vm sun  
create a row of length 21 made of Block of Gold Block of Redstone  
turn right by 45 degrees  
create a row of length 21 made of Block of Gold Block of Redstone
```



⚡ Create a sun by rotating a row of blocks

We use a loop to do 4 lines

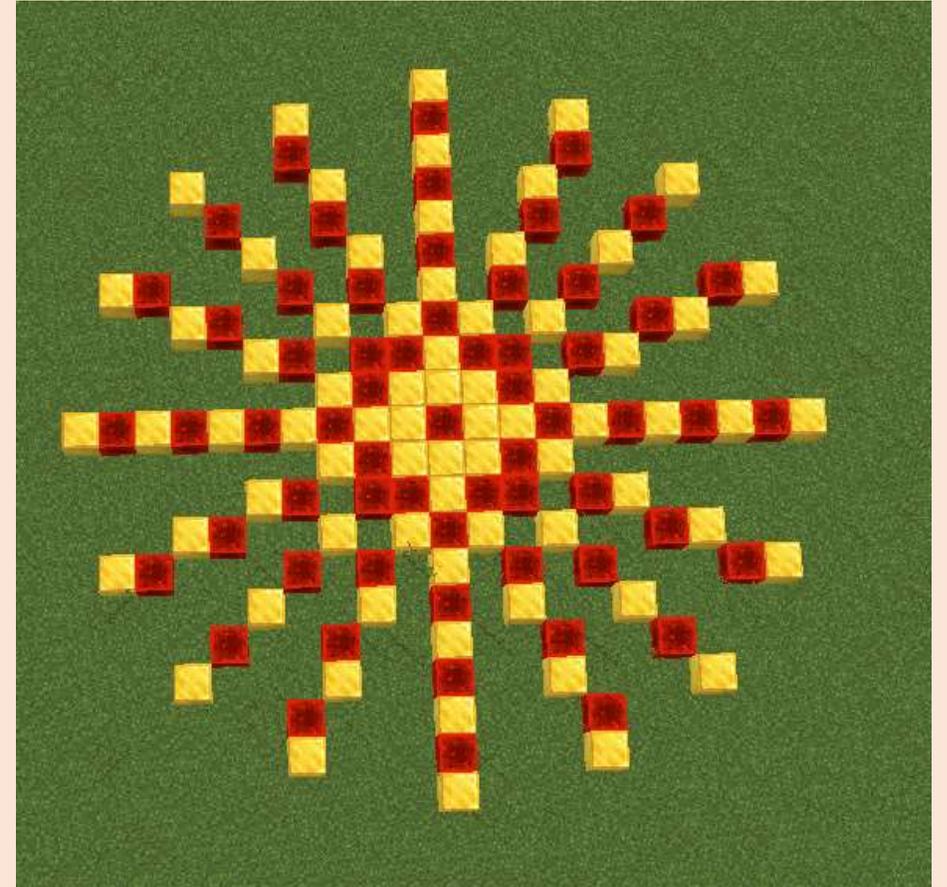
```
vm sun
repeat 4 times
do
  create a row of length 21 made of Block of Gold Block of Redstone
  turn right by 45 degrees
```



How do I double the number of spikes from 4 to 8?

You need to halve the angle and double the repetitions

```
repeat 8 times
do
  create a row of length 21 made of Block of Gold Block of Redstone
  turn right by 22.5 degrees
```

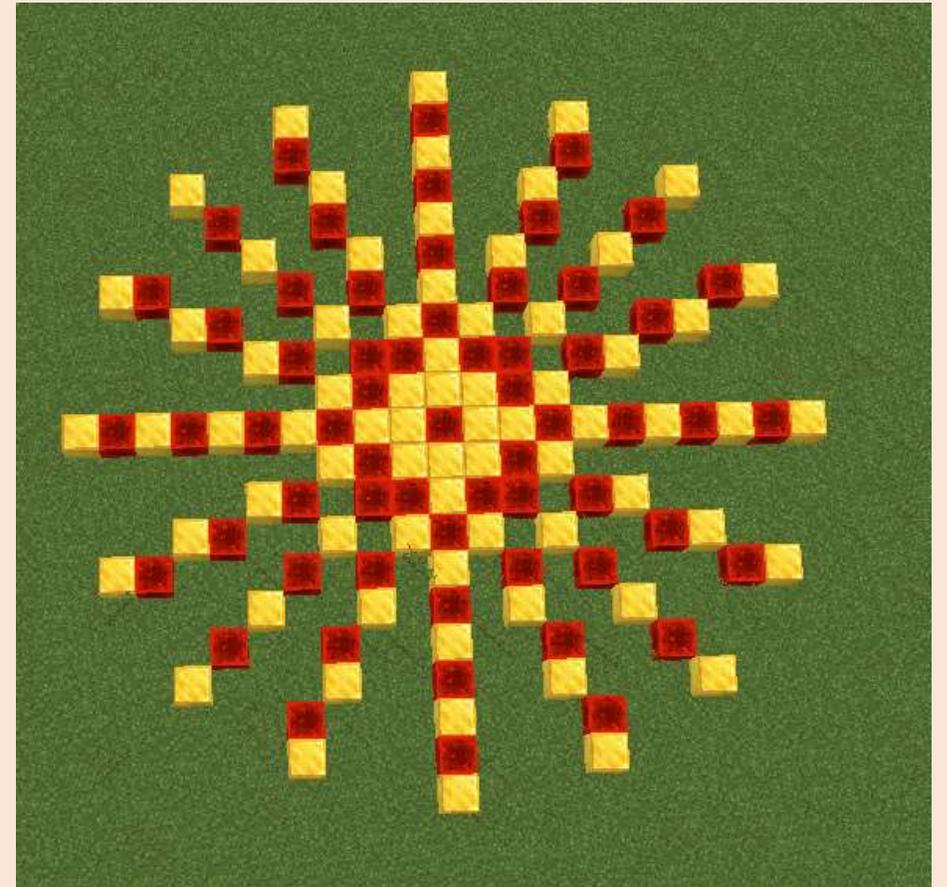


Quiz

How do I double the number of spikes?

Solution

```
repeat 8 times  
do  
  create a row of length 21 made of Block of Gold Block of Redstone  
  turn right by 22.5 degrees
```



Quiz

⚡ Create a rotating stair

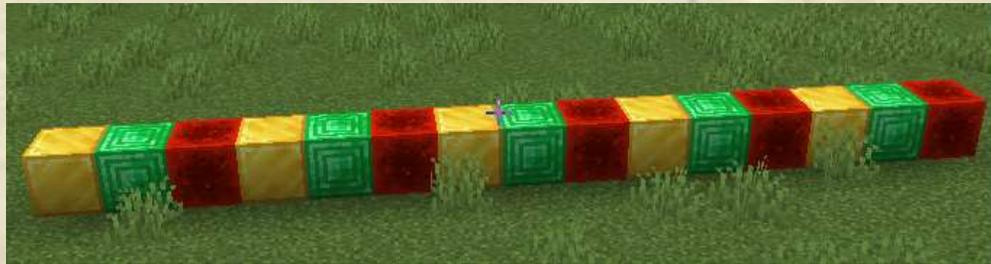
Create a rotating structure by changing the angle and height.



⚡ Create a rotating stair

First we create simple row

```
lvn stair  
create a row of length 15 made of Block of Redstone Block of Emerald Block of Gold
```



⚡ Create a rotating stair

Now we extend the row upwards by using a simple repetition

```
vm stair
repeat 50 times
do
  create a row of length 15 made of Block of Redstone Block of Emerald Block of Gold
  go 1 block up 1
```



⚡ Create a rotating stair

We just add a little rotation of 10 degrees and we have a rotating stair

```
repeat 50 times
do
  create a row of length 15 made of Block of Redstone Block of Emerald Block of Gold
  go 1 block up
  turn right by 10 degrees
```



⚡ The flower thrower

Learn how to set the direction of the robot to where I'm looking.
We are creating a command that generates a row of flowers

```

/vm flowers
repeat 20 times
do
  go 1 block forward ↑
  create a block made of Potted Poppy

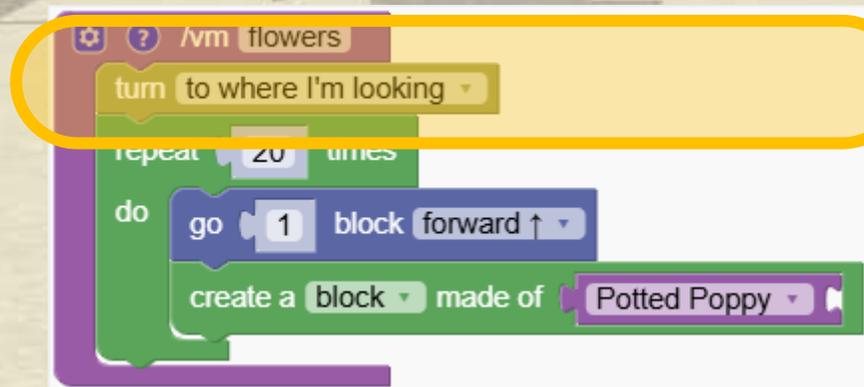
```



⚡ The flower thrower

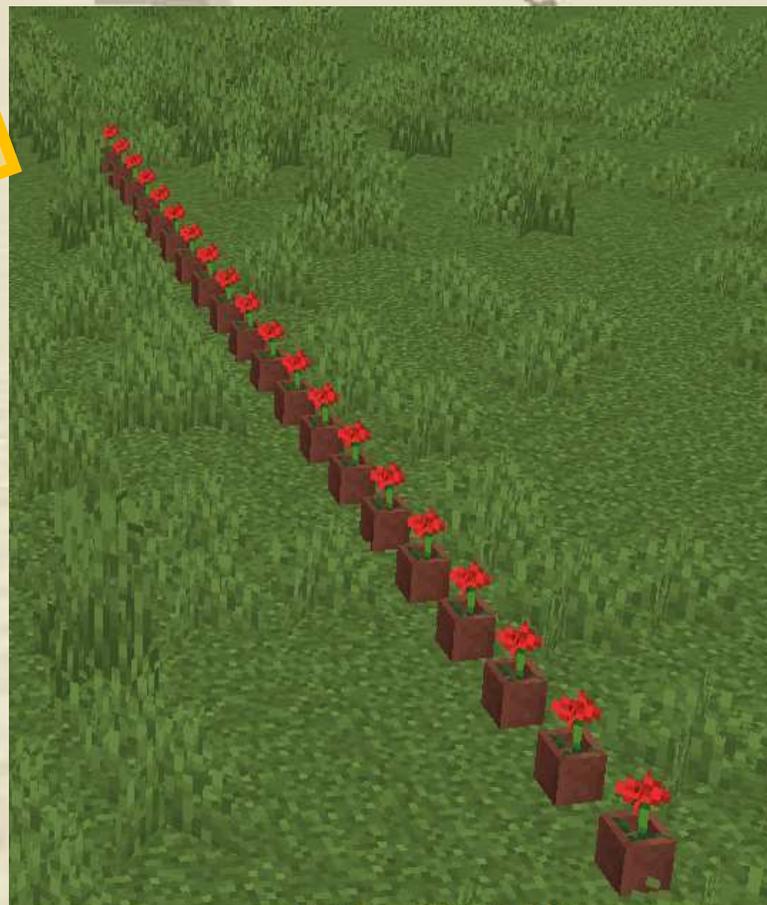
Learn how to set the direction of the robot to where I'm looking.

We are creating a command that generates a row of flowers



```
Scratch code block for 'flowers' command:  
- Command: /vm flowers  
- Action: turn to where I'm looking  
- Loop: repeat 20 times  
- Do block:  
  - Action: go 1 block forward  
  - Action: create a block made of Potted Poppy
```

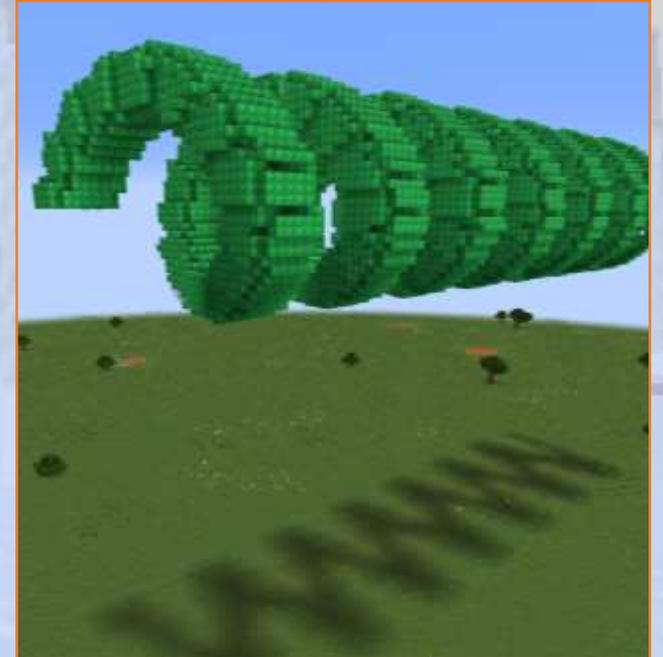
A yellow arrow points from the 'turn to where I'm looking' block to the right.



Vertical Rotation



Amazing structures
created with simple
rotations



Horizontal rotation

 Section Overview

 Objectives

Extend rotations into the vertical dimension to create complex 3D structures

 Expected Outcomes



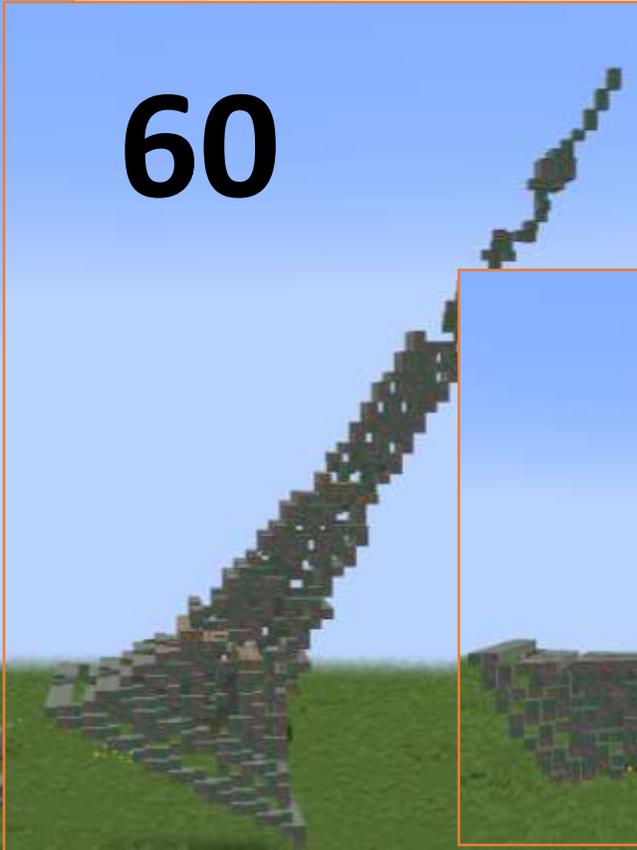
Inclination

Learn how to tilt the robot at different angles.

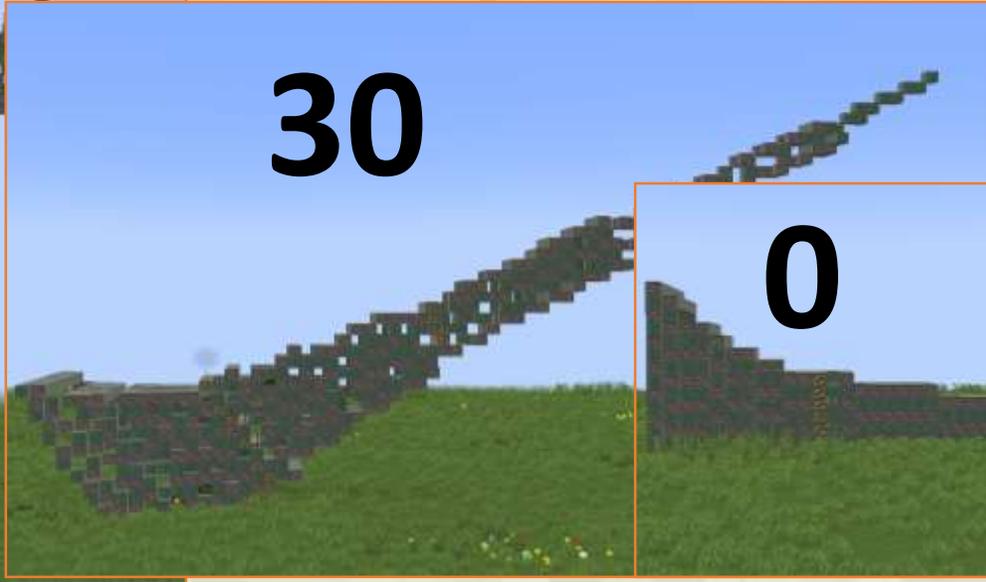
90



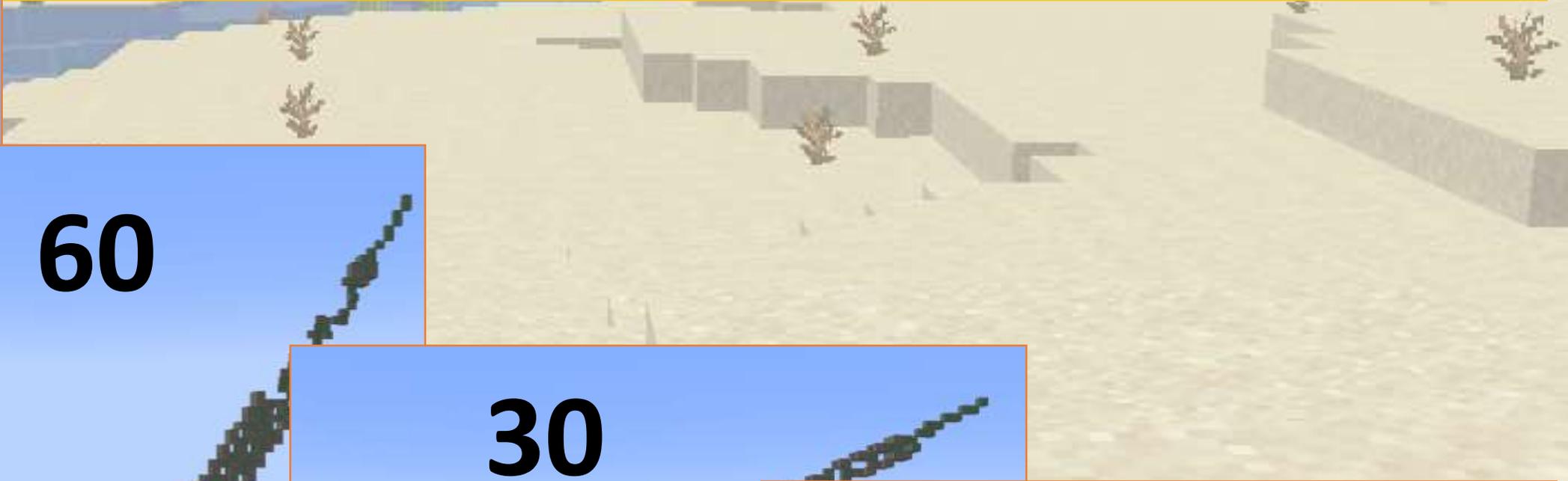
60



30



0



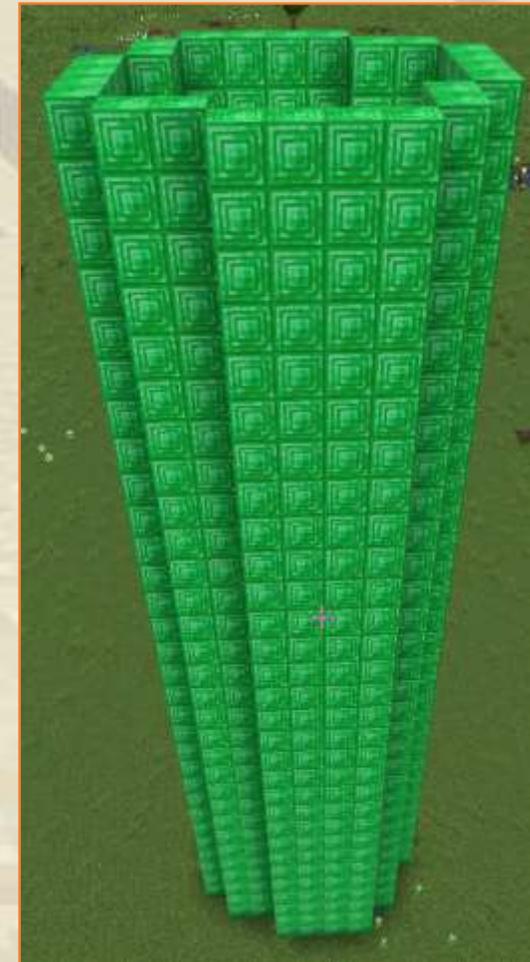
⚡ Tilting towers

Let's have fun with towers. We start with a simple tower

```

/vm tower
repeat 30 times
do
  create a empty circle of radius 4 made of Block of Emerald
  go 1 block up

```



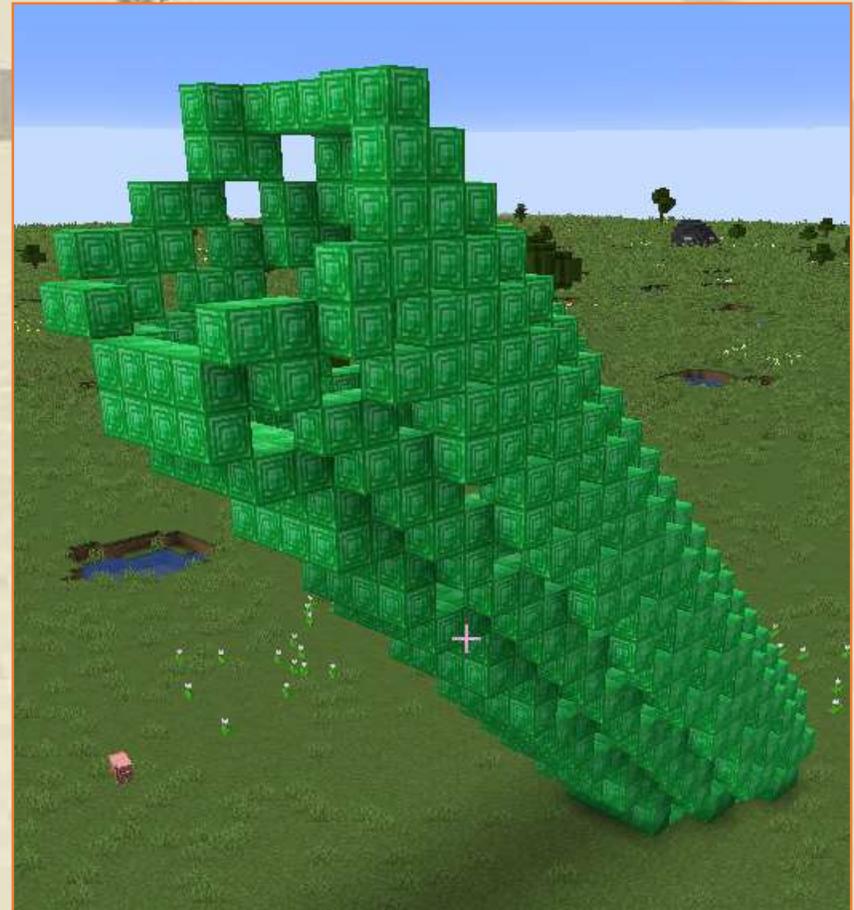
⚡ Tilting towers

Construct a tower with a tilt. Use the block that sets the tilt

```
set tilt to 45 degrees
repeat 30 times
do
  create a empty circle of radius 4 made of Block of Emerald
  go 1 block up
```

```
set tilt to 0 degrees
```

```
change tilt by 45 degrees
```



⚡ Tilting towers

With continuous inclination we manage to curve a tower.
Use the block that changes the tilt inside the repetition.

```
repeat 30 times
do
  create a empty circle of radius 4 made of Block of Emerald
  go 1 block up
  change tilt by 5 degrees
```

```
set tilt to 0 degrees
```

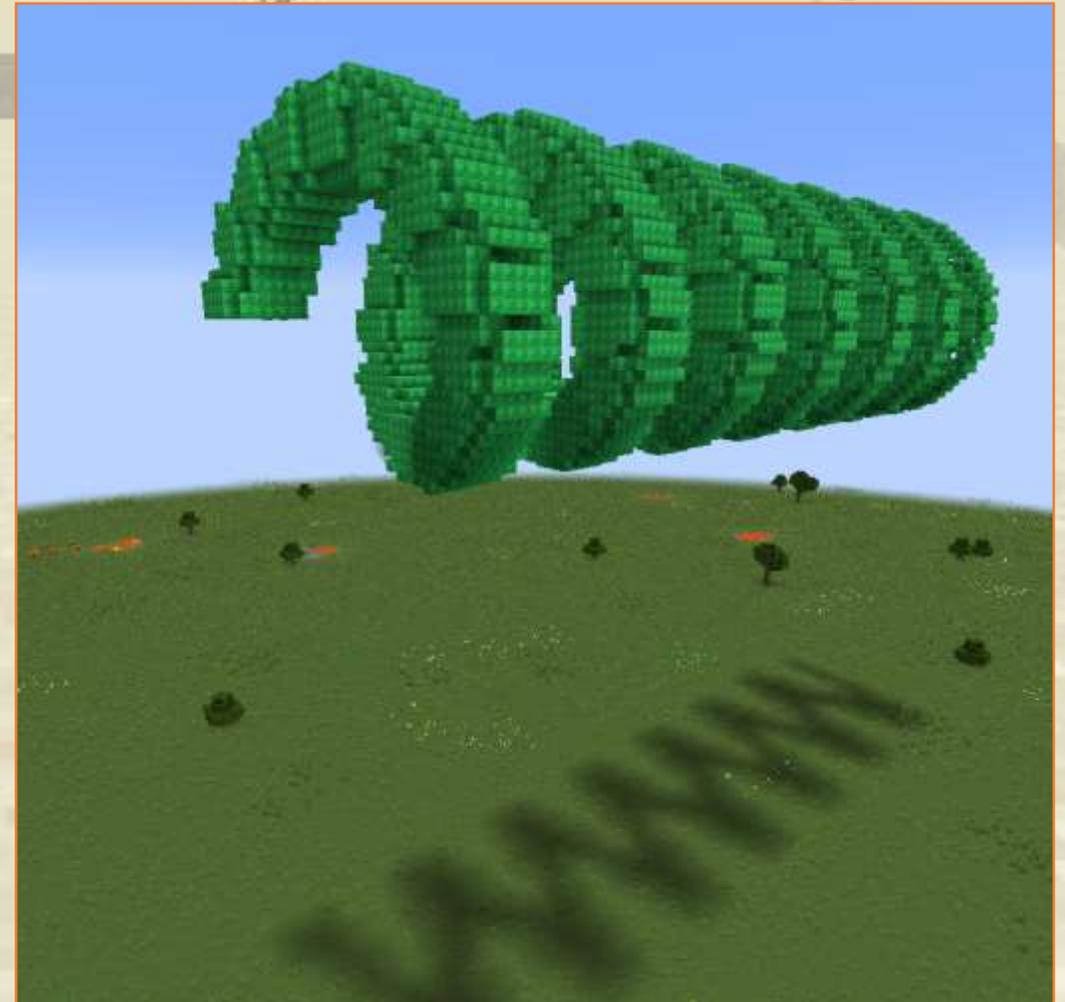
```
change tilt by 45 degrees
```



⚡ Tilting towers

Create a stunning spiraling tower by repeating the curvy tower design. We move slowly sideways “0.2” blocks every time.

```
repeat 300 times
do
  create a empty circle of radius 4 made of Block of Emerald
  go 0.2 block right →
  go 1 block up
  change tilt by 5 degrees
```



⚡ A rainbow in the sky

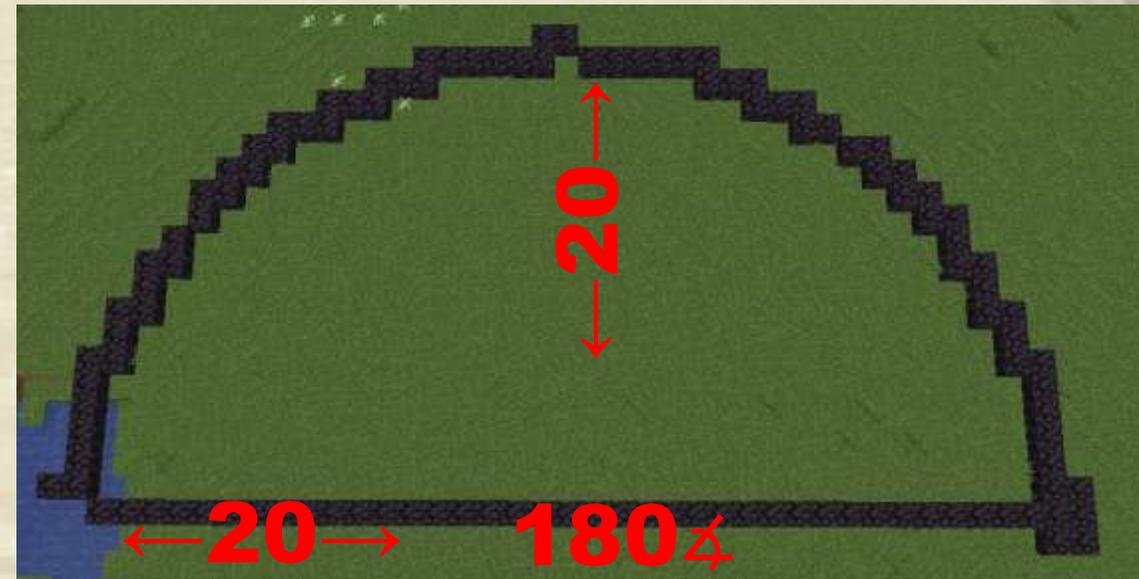
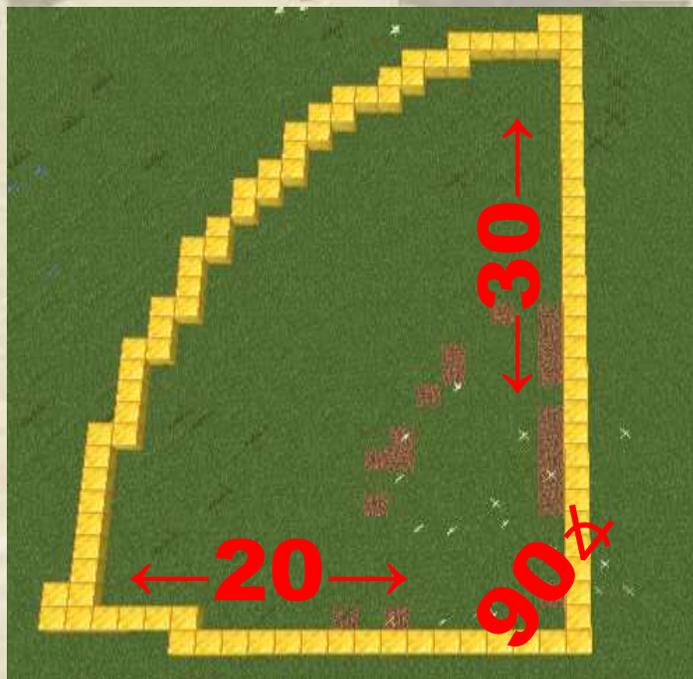
We are creating a rainbow with colored glass



⚡ A rainbow in the sky

The block to create an arc has a width a height and an angle

```
create a empty arc with radius X 20 radius Y 30 and angle 90 made of Block of Gold
```



```
create a empty arc with radius X 20 radius Y 20 and angle 180 made of Blackstone
```

⚡ A rainbow in the sky

We are creating a rainbow with colored glass



⚡ A rainbow in the sky

We are making many arcs. They are full so that no spaces are left behind.

The last smaller arc made of air make sure it looks like an arc



```
create a full arc with radius X 24 radius Y 24 and angle 180 made of Red Stained Glass
create a full arc with radius X 22 radius Y 22 and angle 180 made of Orange Stained Glass
create a full arc with radius X 20 radius Y 20 and angle 180 made of Yellow Stained Glass
create a full arc with radius X 18 radius Y 18 and angle 180 made of Green Stained Glass
create a full arc with radius X 16 radius Y 16 and angle 180 made of Light Blue Stained Glass
create a full arc with radius X 14 radius Y 14 and angle 180 made of Blue Stained Glass
create a full arc with radius X 12 radius Y 12 and angle 180 made of Purple Stained Glass
create a full arc with radius X 10 radius Y 10 and angle 180 made of Air
```



⚡ A rainbow in the sky

To make the arcs vertical we can simple change the tilt

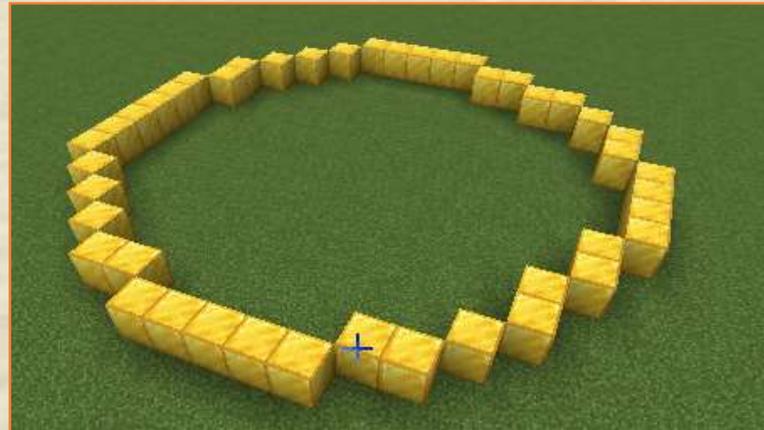
```
set tilt to 90 degrees  
create a full arc with radius X 24 radius Y 24 and angle 180 made of Red Stained Glass  
create a full arc with radius X 22 radius Y 22 and angle 180 made of Orange Stained Glass  
create a full arc with radius X 20 radius Y 20 and angle 180 made of Yellow Stained Glass  
create a full arc with radius X 18 radius Y 18 and angle 180 made of Green Stained Glass  
create a full arc with radius X 16 radius Y 16 and angle 180 made of Light Blue Stained Glass  
create a full arc with radius X 14 radius Y 14 and angle 180 made of Blue Stained Glass  
create a full arc with radius X 12 radius Y 12 and angle 180 made of Purple Stained Glass  
create a full arc with radius X 10 radius Y 10 and angle 180 made of Air
```



⚡ Creating a ball

Design a spherical shape by altering inclination.
We start with a simple circle.

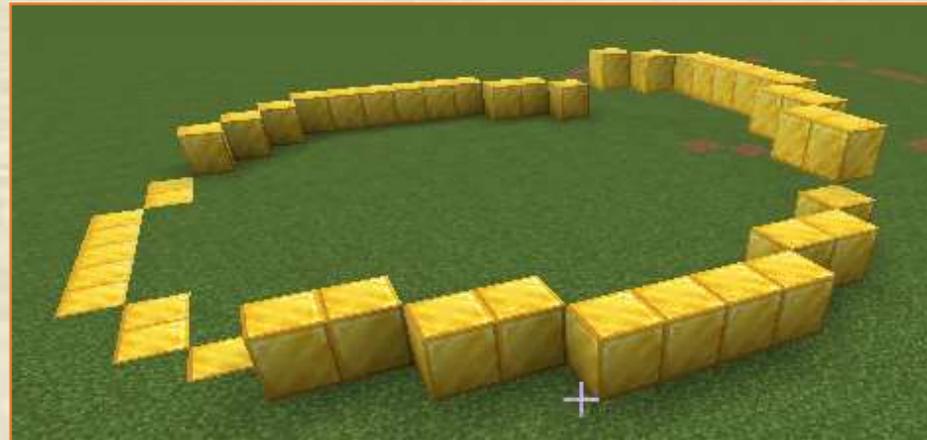
```
vm ball  
create a empty circle of radius 8 made of Block of Gold
```



⚡ Creating a ball

Now the tilt the circle by 5 degrees

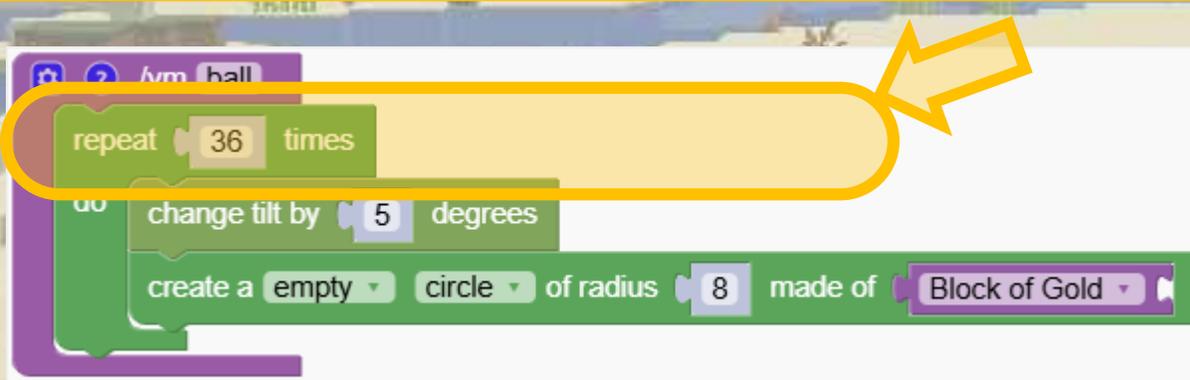
```
vm ball
change tilt by 5 degrees
create a empty circle of radius 8 made of Block of Gold
```



⚡ Creating a ball

Just repeat it 36 times and you have a ball.

Why do we repeat 36 times? Because $180 \text{ degrees} \div 5 \text{ degrees} = 36$



Can you create a cherry?

Combine the tower program with the ball program below

```
vm tower
repeat 30 times
do
  create a empty circle of radius 4 made of Green Wool
  go 1 block up ↑
  change tilt by 5 degrees
```

```
vm ball
repeat 36 times
do
  change tilt by 5 degrees
  create a empty circle of radius 16 made of Red Wool
```



Quiz

Can you create a cherry?

Combine the tower program with the ball program below

```

/vm cherry
go 20 block forward ↑
repeat 36 times
do
change tilt by 5 degrees
create a empty circle of radius 16 made of Red Wool
set tilt to 0 degrees
go 16 block up ↑
repeat 30 times
do
create a empty circle of radius 4 made of Green Wool
go 1 block up ↑
change tilt by 5 degrees

```

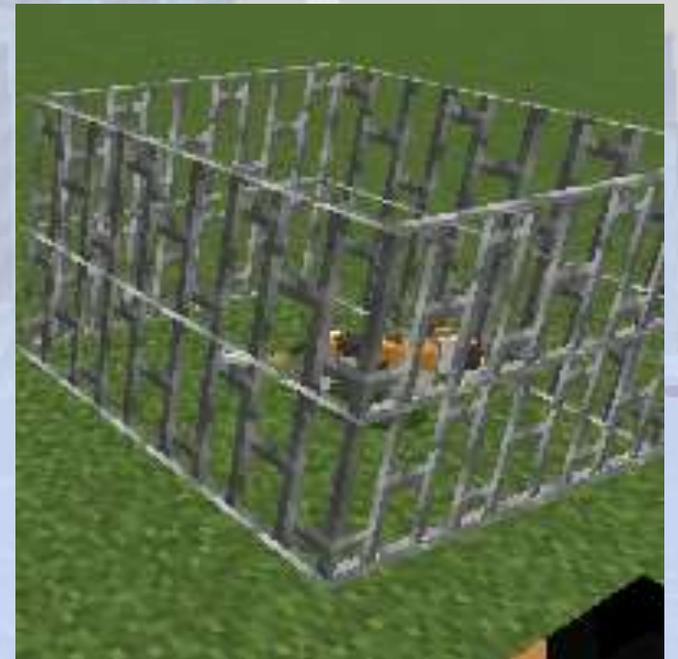


Quiz

Functions



Organize code into
functions



Functions

 Section Overview

 Objectives

Organize code into reusable functions to make it easier to understand and maintain.

 Expected Outcomes

What are we going to learn

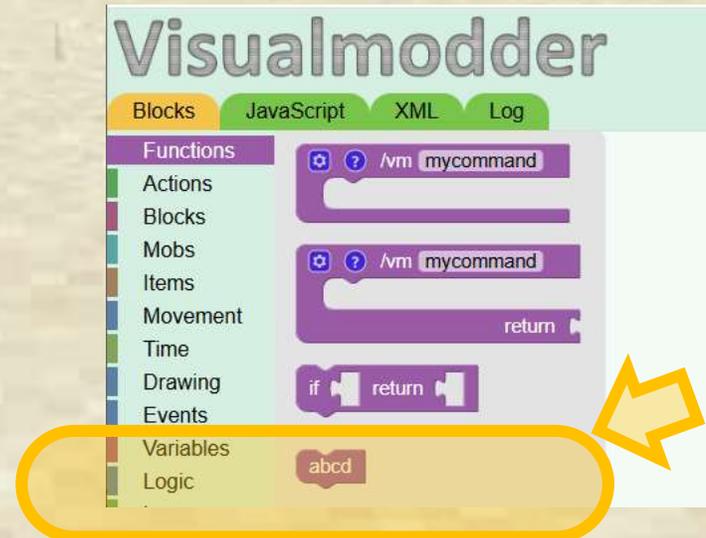
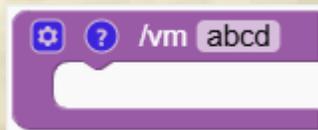
We learn how to use functions to organize code, reduce repetition, and improve readability.



Purpose of defining functions

Learn the basics of creating and using functions in code.

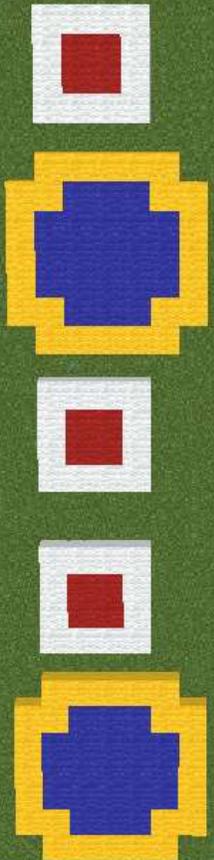
When we create a new function “abcd”, in the menu we find a block representing the new function “abcd”



Functions organize our code

Understand how functions simplify complex code.

We are creating this picture and we can write our code like this or..



```
 /vm pic2
create a full square of width 4 made of Red Wool
create a empty square of width 4 made of White Wool
go 6 block forward ↑
create a full circle of radius 4 made of Blue Wool
create a empty circle of radius 4 made of Yellow Wool
go 7 block forward ↑
create a full square of width 4 made of Red Wool
create a empty square of width 4 made of White Wool
go 6 block forward ↑
create a full square of width 4 made of Red Wool
create a empty square of width 4 made of White Wool
go 6 block forward ↑
create a full circle of radius 4 made of Blue Wool
create a empty circle of radius 4 made of Yellow Wool
go 7 block forward ↑
```

Functions organize our code

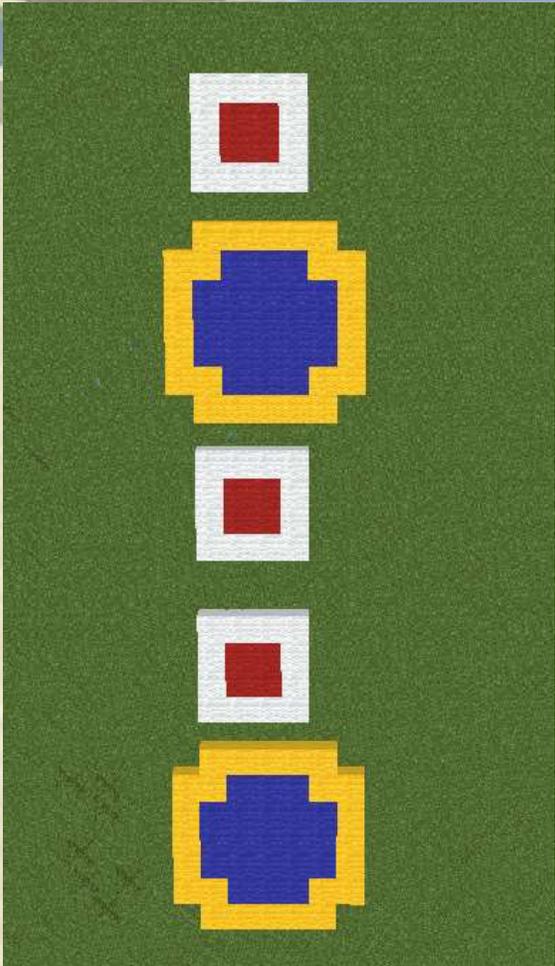
Understand how functions simplify complex code.

Like this !

```
/vm pic  
square  
circle  
square  
square  
circle
```

```
/vm circle  
create a full circle of radius 4 made of Blue Wool  
create a empty circle of radius 4 made of Yellow Wool  
go 7 block forward
```

```
/vm square  
create a full square of width 4 made of Red Wool  
create a empty square of width 4 made of White Wool  
go 6 block forward
```



Functions organize our code

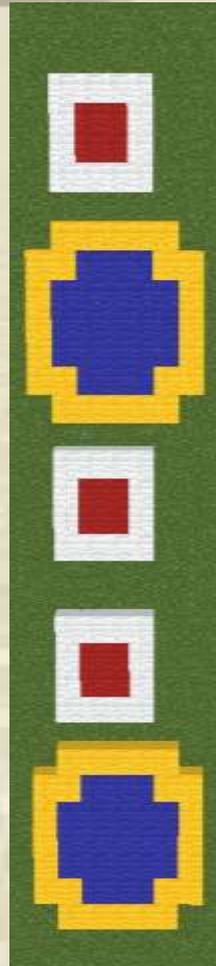
Simplify complex code.

```
function /vm pic
  square
  circle
  square
  square
  circle
```

This code is easier to understand

```
function /vm circle
  create a full circle of radius 4 made of Blue Wool
  create a empty circle of radius 4 made of Yellow Wool
  go 7 block forward
```

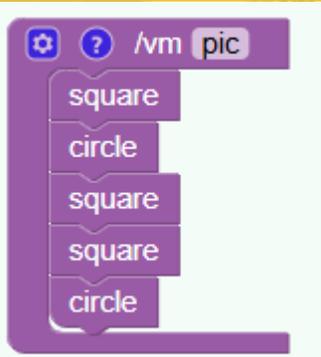
```
function /vm square
  create a full square of width 4 made of Red Wool
  create a empty square of width 4 made of White Wool
  go 6 block forward
```



```
function /vm pic2
  create a full square of width 4 made of Red Wool
  create a empty square of width 4 made of White Wool
  go 6 block forward
  create a full circle of radius 4 made of Blue Wool
  create a empty circle of radius 4 made of Yellow Wool
  go 7 block forward
  create a full square of width 4 made of Red Wool
  create a empty square of width 4 made of White Wool
  go 6 block forward
  create a full circle of radius 4 made of Blue Wool
  create a empty circle of radius 4 made of Yellow Wool
  go 7 block forward
```

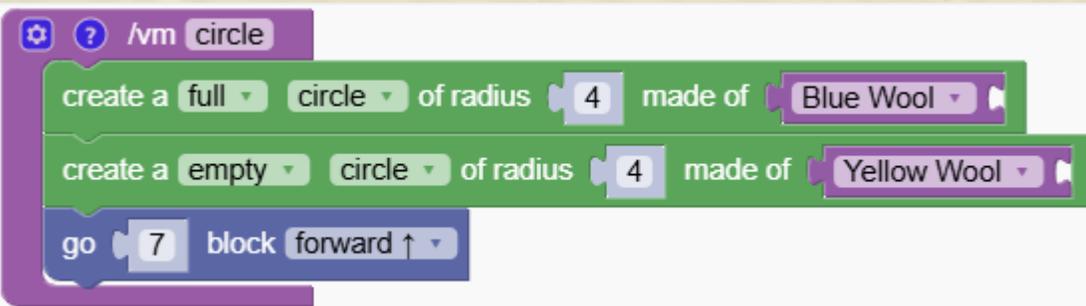
Functions organize our code

Avoid repeating code

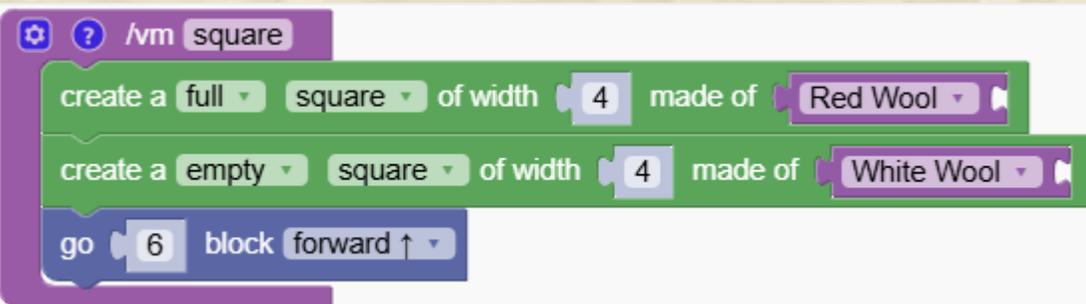


```
function pic {  
  square  
  circle  
  square  
  square  
  circle  
}
```

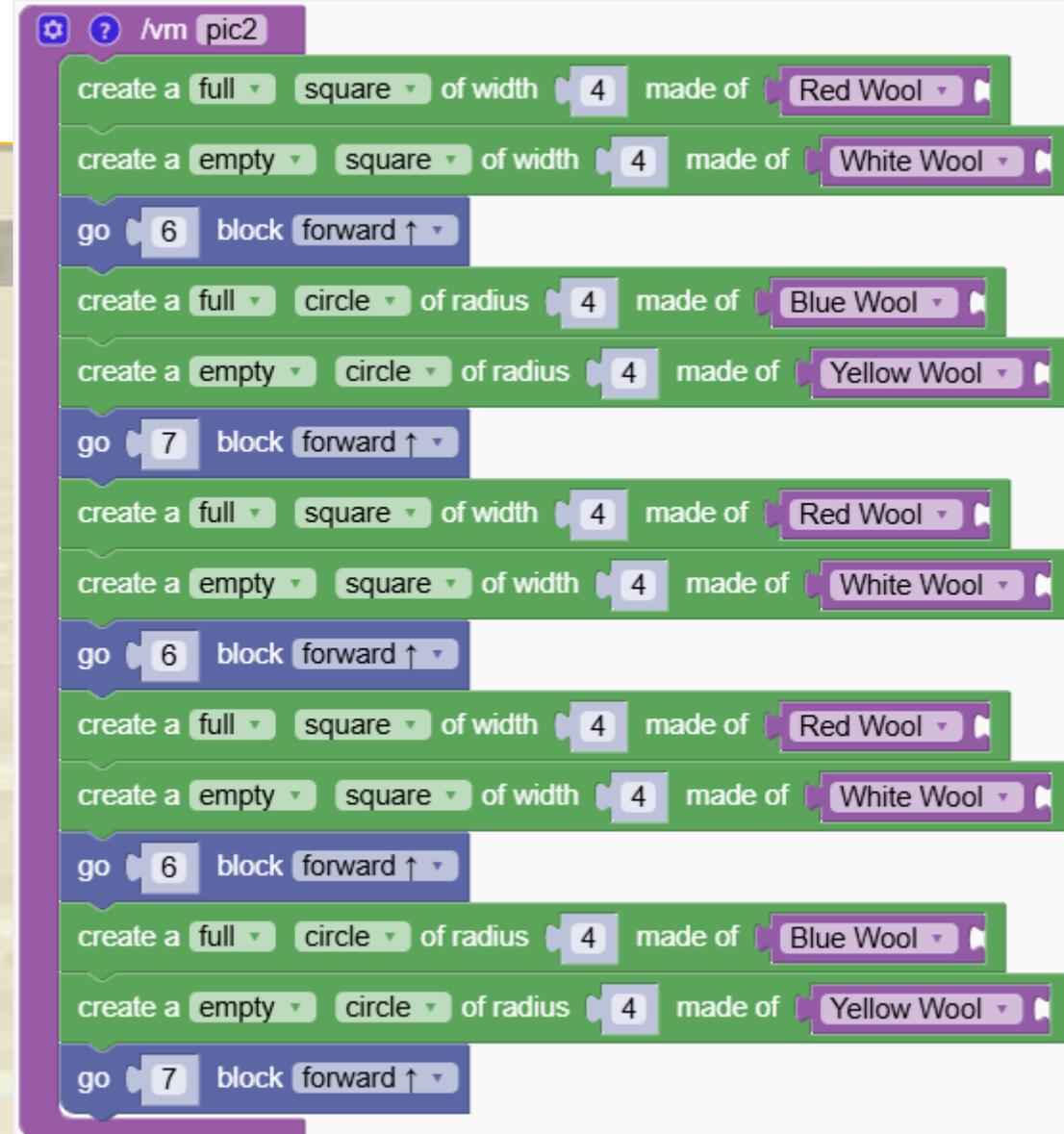
If I want to change the color of the squares I do it only in one place



```
function circle {  
  create a full circle of radius 4 made of Blue Wool  
  create an empty circle of radius 4 made of Yellow Wool  
  go 7 block forward  
}
```



```
function square {  
  create a full square of width 4 made of Red Wool  
  create an empty square of width 4 made of White Wool  
  go 6 block forward  
}
```



```
function pic2 {  
  pic  
  create a full square of width 4 made of Red Wool  
  create an empty square of width 4 made of White Wool  
  go 6 block forward  
  create a full circle of radius 4 made of Blue Wool  
  create an empty circle of radius 4 made of Yellow Wool  
  go 7 block forward  
  create a full square of width 4 made of Red Wool  
  create an empty square of width 4 made of White Wool  
  go 6 block forward  
  create a full circle of radius 4 made of Blue Wool  
  create an empty circle of radius 4 made of Yellow Wool  
  go 7 block forward  
}
```

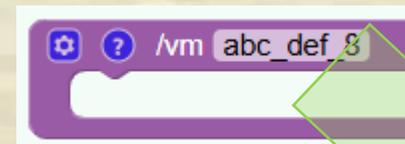
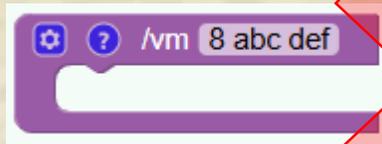
Functions naming

Learn how to properly name functions to avoid errors.

Function names:

- 1: Must start with a letter or the characters – and _
- 2: Can use letters, digits or the characters – and _

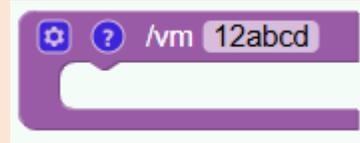
The function below has no valid name because it starts with a number and contains spaces. It should be rewritten using _ and the number can be moved to the end.



Which function names are valid?

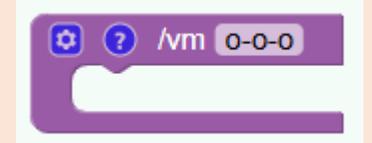
Look at the examples and decide which function names are valid

?



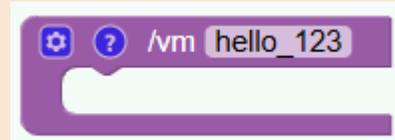
```
/vm 12abcd
```

?



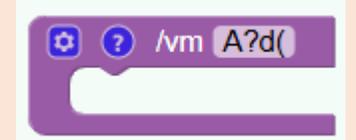
```
/vm 0-0-0
```

?



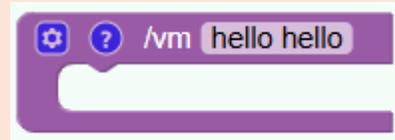
```
/vm hello_123
```

?



```
/vm A?d(
```

?



```
/vm hello hello
```

?



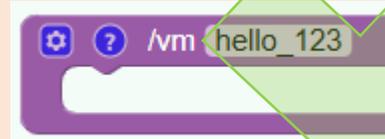
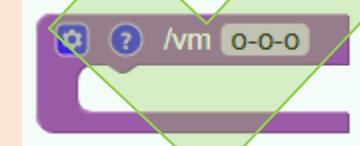
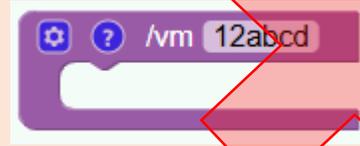
```
/vm :-)
```

Quiz

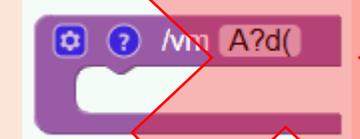
Which function names are valid?

Solution:

Starts with a digit



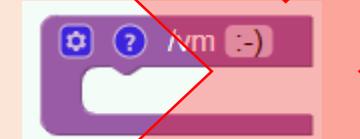
Contains invalid characters



Contains a space



Contains invalid characters



Quiz

⚡ Castle with towers

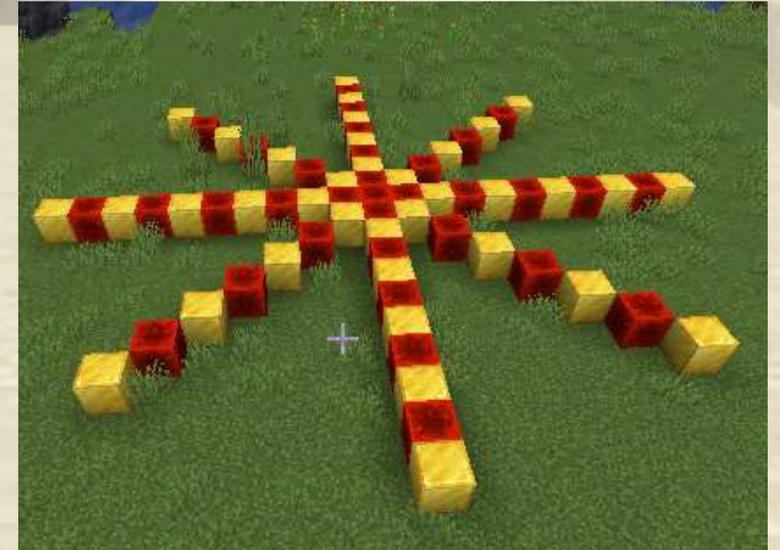
Create a beautiful castle surrounded by towers.



⚡ Castle with towers

Previously we saw this program.
Now we can adapt it to generate an amazing castle

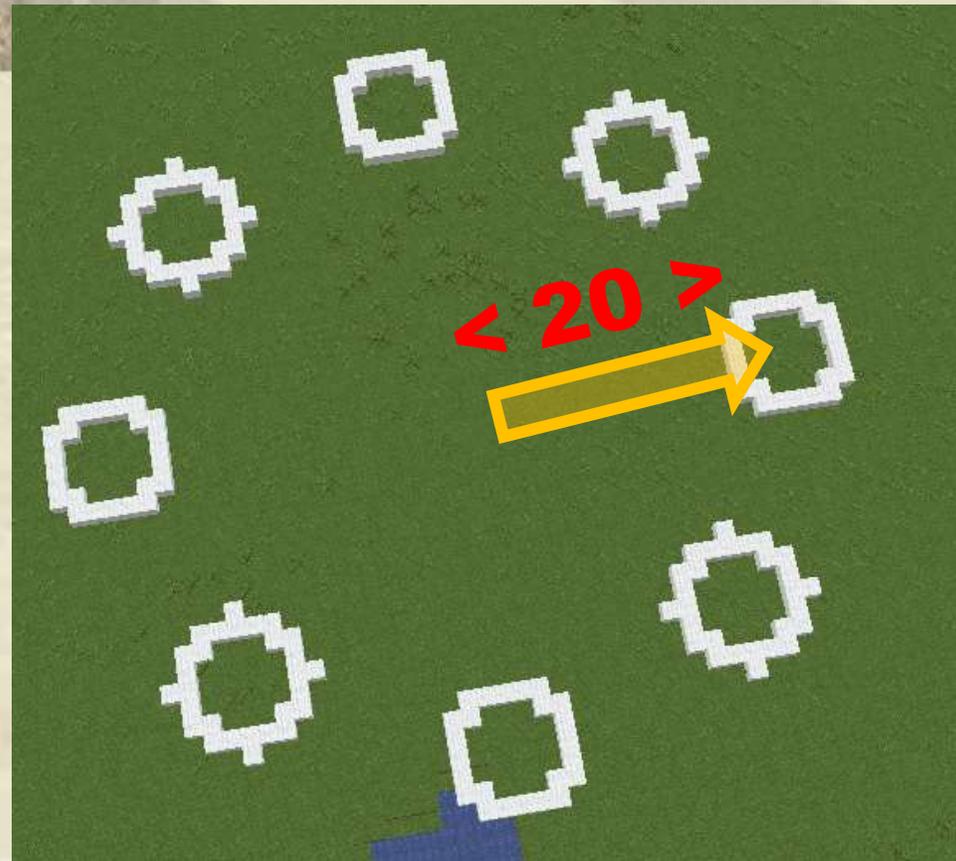
```
vm sun
repeat 4 times
do
  create a row of length 21 made of Block of Gold Block of Redstone
  turn right by 45 degrees
```



⚡ Castle with towers

This program is similar to the previous one. Instead of creating a line of blocks, we put a circle.

```
function castle
repeat 8 times
do
  go 20 block forward
  create a empty circle of radius 4 made of White Wool
  go to the start
  turn right by 45 degrees
```

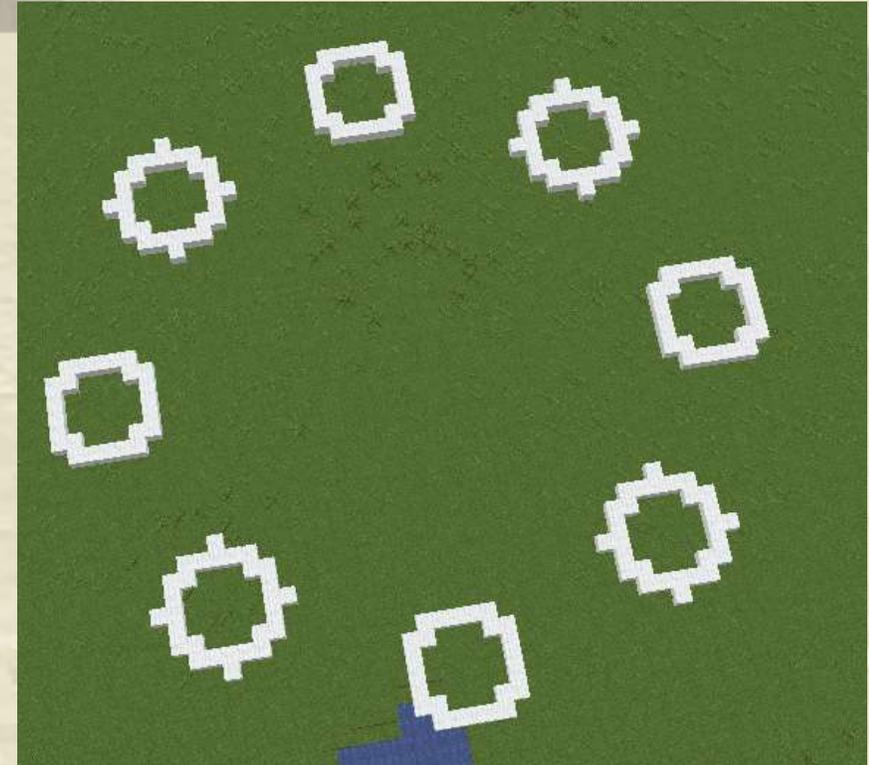


⚡ Castle with towers

We clean up our code by creating a function “tower1”.
The program makes the same circles as before

```
vm castle
repeat 8 times
do
tower1
turn right by 45 degrees
```

```
vm tower1
go 20 block forward
create a empty circle of radius 4 made of White Wool
go to the start
```

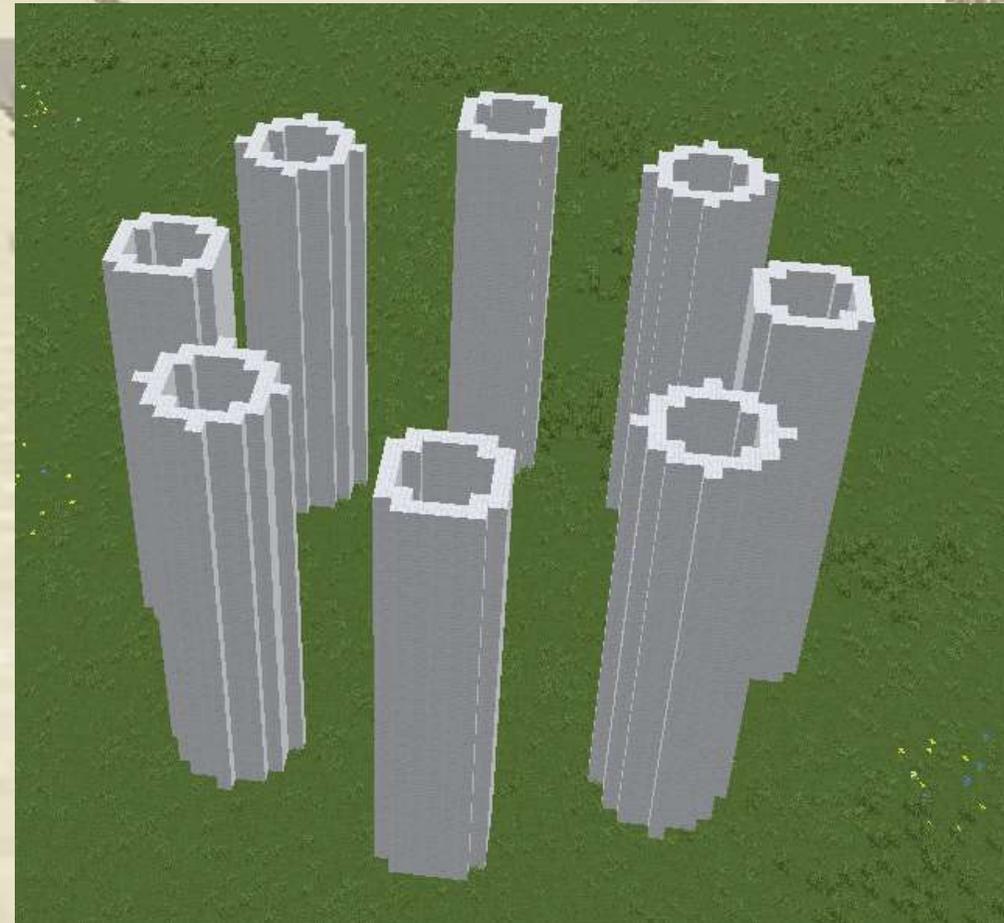


⚡ Castle with towers

Instead of making a simple circle now we create a tower.

```
vm castle
repeat 8 times
do
  tower1
  turn right by 45 degrees
```

```
vm tower1
go 20 block forward
repeat 30 times
do
  create a empty circle of radius 4 made of White Wool
  go 1 block up
go to the start
```



⚡ Castle with towers

We cloned the “tower1” function and compressed it

Now we have a new function “tower2” and we call it every time together with “tower1”

```
script /vm castle
repeat 8 times
do
tower1
tower2
turn right by 45 degrees
```

```
script /vm tower1 go 20 block for...
```

```
script tower2
go 30 block forward
repeat 20 times
do
create a empty circle of radius 4 made of Red Wood
go 1 block up
go to the start
```



⚡ Castle with towers

We repeated the previous step and now we have 3 functions for the towers.

```

/vm castle
repeat 8 times
do
  tower1
  tower2
  tower3
  turn right by 45 degrees

```

```

/vm tower1 go 20 block for...

```

```

/vm tower3 go 40 block for...

```

```

/vm tower3
go 40 block forward
repeat 10 times
do
  create a empty circle of radius 4 made of Yellow Wood
  go 1 block up
go to the start

```



⚡ Castle with towers

To finish, we made a function “walls” that creates 3 walls using simple circles

```
vm castle
walls
repeat 8 times
do
tower1
tower2
tower3
turn right by 45 degrees
```

```
/vm tower1 go 20 block for...
```

```
/vm tower2 go 30 block for...
```

```
/vm tower3 go 40 block for...
```

```
vm walls
repeat 8 times
do
create a empty circle of radius 20 made of White Wool
create a empty circle of radius 30 made of Red Wool
create a empty circle of radius 40 made of Yellow Wool
go 1 block up
go to the start
```



Make your own circles and towers

Experiment with creating your own castles.



⚡ Programmable potions

Learn how functions can be used to create programmable potions.

First we create a simple function that creates a cage

```
function /vm cage  
  create a empty square of width 4 made of Iron Bars  
  go 1 blocks up  
  create a empty square of width 4 made of Iron Bars
```



⚡ Programmable potions

Now we create a second command that gives us a potion that, when thrown, calls the previous function

```
/vm cage  
create a empty square of width 4 made of Iron Bars  
go 1 blocks up 1  
create a empty square of width 4 made of Iron Bars
```

```
/vm trap  
give me splash potion with function cage
```



Put here the name of the function you want to call

Fun ⚡ Catch each other in Minecraft

Enjoy a group activity that involves catching each other using your programmed potions.

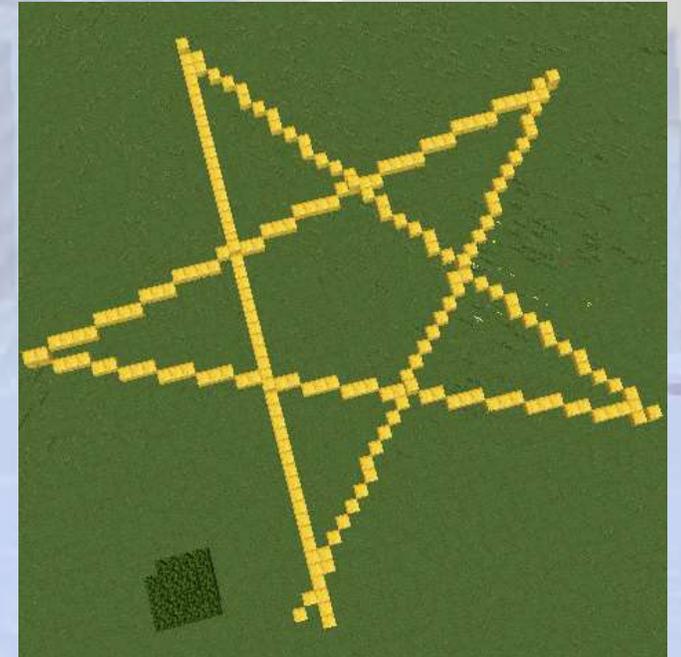
Customize the function so that they do different actions, like making mobs appear or build instant towers. There are no limits to your ideas!



Advanced Positioning



Learn how to mark
specific positions



Advanced Positioning

Section Overview

We explore marking positions and leveraging them to build more complex designs and patterns.

After many movements it can be difficult to figure out how to go back to a specific position. Marking a specific position is very useful and saves us effort to calculate correctly how to go back to a point

Objectives

Learn how to mark specific positions and use them to create advanced structures.

Expected Outcomes

Students will be able to build complex structures where marking a position is important

Marking Positions

Follow this example to understand the concept and importance of marking positions.



```
comando /vm  
crea un blocco fatto di Blocco di neve  
vai 5 blocchi in avanti ↑  
crea un blocco fatto di Blocco d'oro  
vai 4 blocchi a destra →  
crea un blocco fatto di Blocco di bambù  
vai alla partenza  
vai 7 blocchi a sinistra ←  
crea un blocco fatto di Alveare  
marca questo blocco  
vai 3 blocchi in dietro ↓  
crea un blocco fatto di Blocco di diamante  
vai 5 blocchi a destra →  
crea un blocco fatto di Blocco di rame  
vai alla posizione marcata  
vai 1 blocchi a destra →  
crea un blocco fatto di Assi di betulla  
vai alla partenza  
vai 1 blocchi in su ↑  
crea un blocco fatto di Lanterna di zucca
```

Marking Positions

Our starting position is at the snow block



```
crea un blocco fatto di Blocco di neve
vai 5 blocchi in avanti ↑
crea un blocco fatto di Blocco d'oro
vai 4 blocchi a destra →
crea un blocco fatto di Blocco di bambù
vai alla partenza
vai 7 blocchi a sinistra ←
crea un blocco fatto di Alveare
marca questo blocco
vai 3 blocchi in dietro ↓
crea un blocco fatto di Blocco di diamante
vai 5 blocchi a destra →
crea un blocco fatto di Blocco di rame
vai alla posizione marcata
vai 1 blocchi a destra →
crea un blocco fatto di Assi di betulla
vai alla partenza
vai 1 blocchi in su ↑
crea un blocco fatto di Lanterna di zucca
```

Marking Positions

We move forward and create a block of gold



```
comando /vm  
crea un blocco fatto di Blocco di neve  
vai 5 blocchi in avanti ↑  
crea un blocco fatto di Blocco d'oro  
vai 4 blocchi a destra →  
crea un blocco fatto di Blocco di bambù  
vai alla partenza  
vai 7 blocchi a sinistra ←  
crea un blocco fatto di Alveare  
marca questo blocco  
vai 3 blocchi in dietro ↓  
crea un blocco fatto di Blocco di diamante  
vai 5 blocchi a destra →  
crea un blocco fatto di Blocco di rame  
vai alla posizione marcata  
vai 1 blocchi a destra →  
crea un blocco fatto di Assi di betulla  
vai alla partenza  
vai 1 blocchi in su ↑  
crea un blocco fatto di Lanterna di zucca
```

Marking Positions

We move right and create the next block



```
comando /vm  
crea un blocco fatto di Blocco di neve  
vai 5 blocchi in avanti ↑  
crea un blocco fatto di Blocco d'oro  
vai 4 blocchi a destra →  
crea un blocco fatto di Blocco di bambù  
vai alla partenza  
vai 7 blocchi a sinistra ←  
crea un blocco fatto di Alveare  
marca questo blocco  
vai 3 blocchi in dietro ↓  
crea un blocco fatto di Blocco di diamante  
vai 5 blocchi a destra →  
crea un blocco fatto di Blocco di rame  
vai alla posizione marcata  
vai 1 blocchi a destra →  
crea un blocco fatto di Assi di betulla  
vai alla partenza  
vai 1 blocchi in su ↑  
crea un blocco fatto di Lanterna di zucca
```

Marking Positions

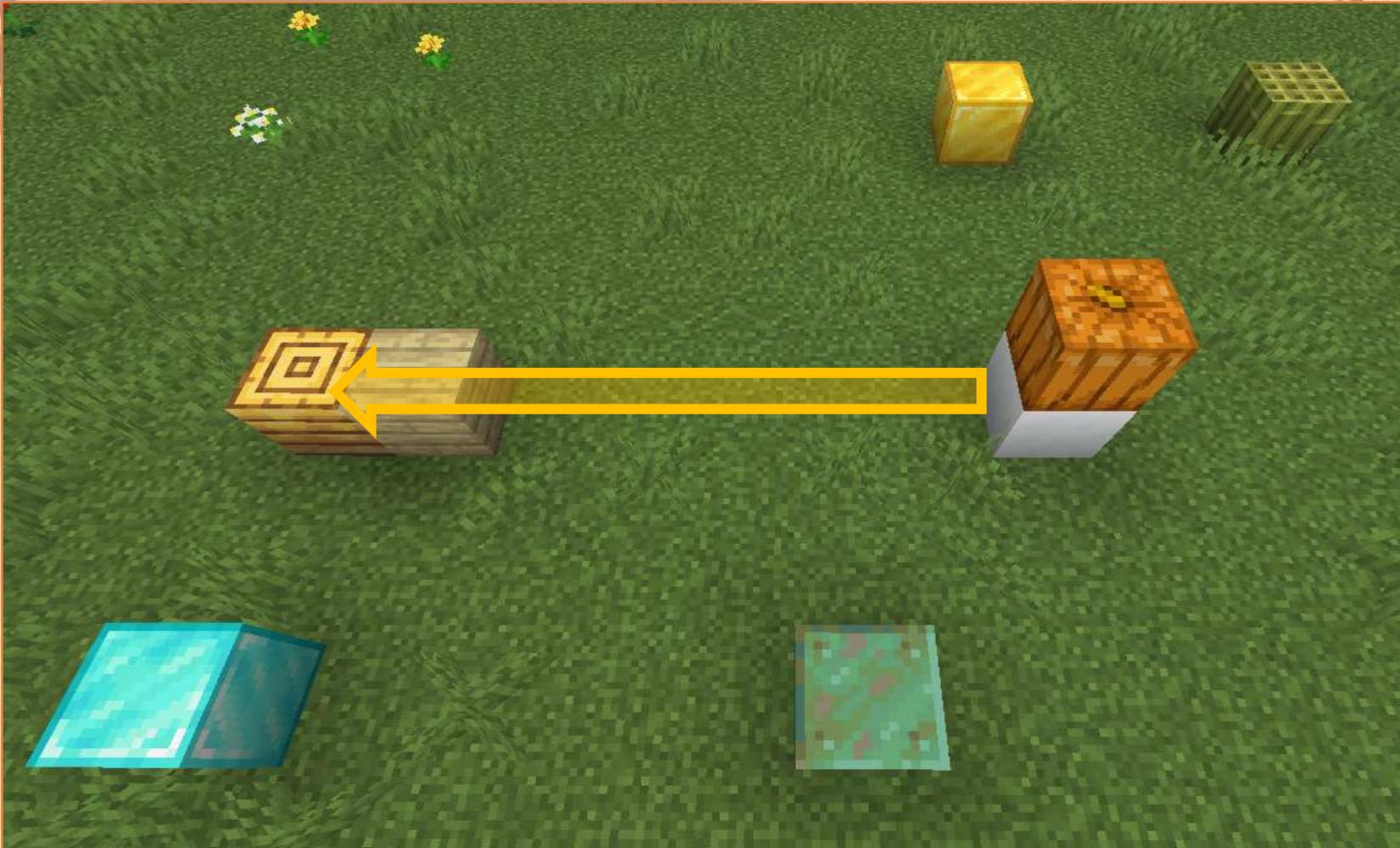
We are moving be to the starting point where we created the block of snow



```
comando /vm  
crea un blocco fatto di Blocco di neve  
vai 5 blocchi in avanti ↑  
crea un blocco fatto di Blocco d'oro  
vai 4 blocchi a destra →  
crea un blocco fatto di Blocco di bambù  
vai alla partenza  
vai 7 blocchi a sinistra ←  
crea un blocco fatto di Alveare  
marca questo blocco  
vai 3 blocchi in dietro ↓  
crea un blocco fatto di Blocco di diamante  
vai 5 blocchi a destra →  
crea un blocco fatto di Blocco di rame  
vai alla posizione marcata  
vai 1 blocchi a destra →  
crea un blocco fatto di Assi di betulla  
vai alla partenza  
vai 1 blocchi in su ↑  
crea un blocco fatto di Lanterna di zucca
```

Marking Positions

We move left



```
comando /vm  
crea un blocco fatto di Blocco di neve  
vai 5 blocchi in avanti ↑  
crea un blocco fatto di Blocco d'oro  
vai 4 blocchi a destra →  
crea un blocco fatto di Blocco di bambù  
vai alla partenza  
vai 7 blocchi a sinistra ←  
crea un blocco fatto di Alveare  
marca questo blocco  
vai 3 blocchi in dietro ↓  
crea un blocco fatto di Blocco di diamante  
vai 5 blocchi a destra →  
crea un blocco fatto di Blocco di rame  
vai alla posizione marcata  
vai 1 blocchi a destra →  
crea un blocco fatto di Assi di betulla  
vai alla partenza  
vai 1 blocchi in su ↑  
crea un blocco fatto di Lanterna di zucca
```

Marking Positions

We tell robot to remember where we are. The star is just in this picture to explain the marking concept.



```
comando
crea un blocco fatto di Blocco di neve
vai 5 blocchi in avanti ↑
crea un blocco fatto di Blocco d'oro
vai 4 blocchi a destra →
crea un blocco fatto di Blocco di bambù
vai alla partenza
vai 7 blocchi a sinistra ←
crea un blocco fatto di Alveare
marca questo blocco
vai 3 blocchi in dietro ↓
crea un blocco fatto di Blocco di diamante
vai 5 blocchi a destra →
crea un blocco fatto di Blocco di rame
vai alla posizione marcata
vai 1 blocchi a destra →
crea un blocco fatto di Assi di betulla
vai alla partenza
vai 1 blocchi in su ↑
crea un blocco fatto di Lanterna di zucca
```

Marking Positions

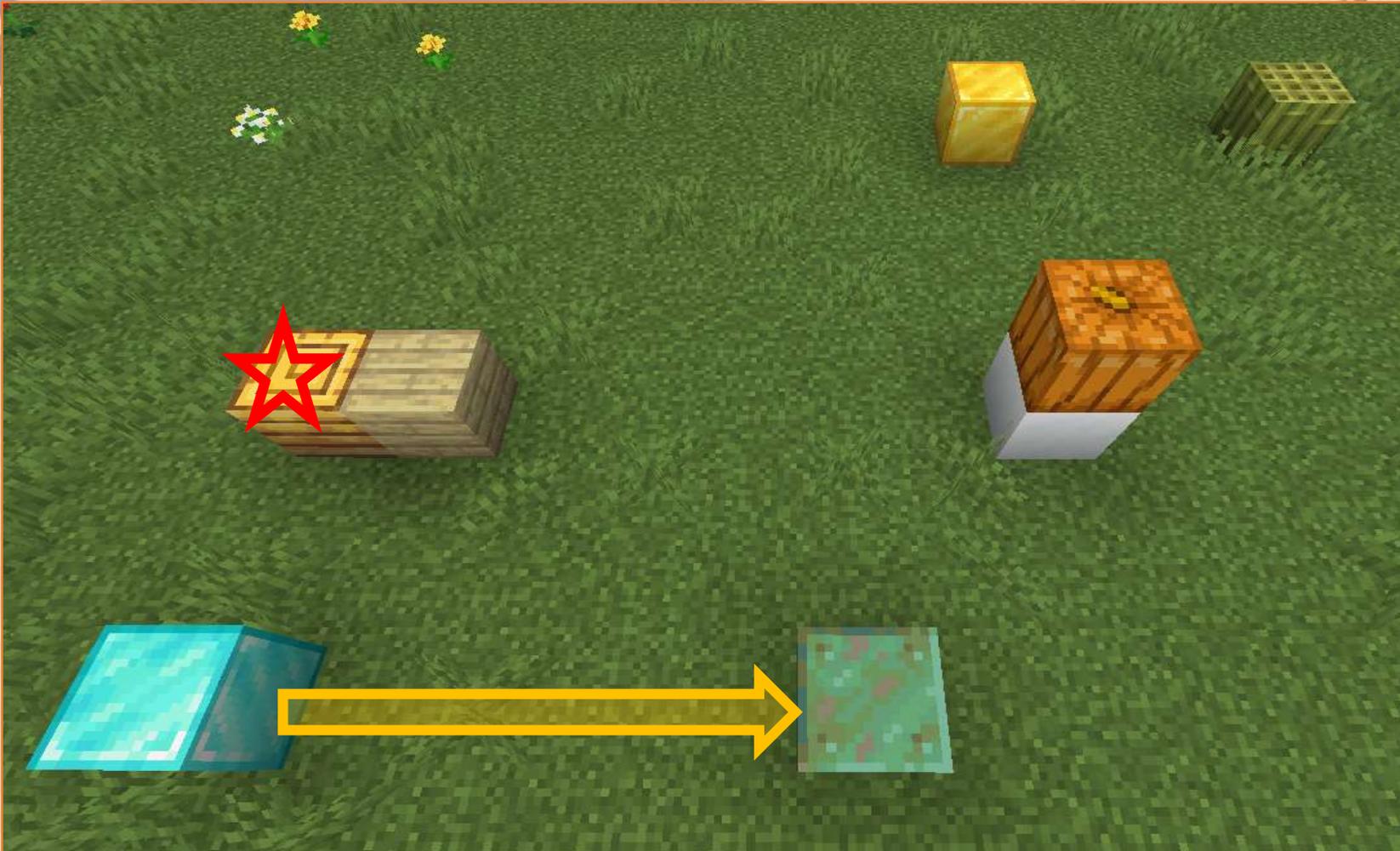
We go back 3 steps



```
comando /vm  
crea un blocco fatto di Blocco di neve  
vai 5 blocchi in avanti ↑  
crea un blocco fatto di Blocco d'oro  
vai 4 blocchi a destra →  
crea un blocco fatto di Blocco di bambù  
vai alla partenza  
vai 7 blocchi a sinistra ←  
crea un blocco fatto di Alveare  
marca questo blocco  
vai 3 blocchi in dietro ↓  
crea un blocco fatto di Blocco di diamante  
vai 5 blocchi a destra →  
crea un blocco fatto di Blocco di rame  
vai alla posizione marcata  
vai 1 blocchi a destra →  
crea un blocco fatto di Assi di betulla  
vai alla partenza  
vai 1 blocchi in su ↑  
crea un blocco fatto di Lanterna di zucca
```

Marking Positions

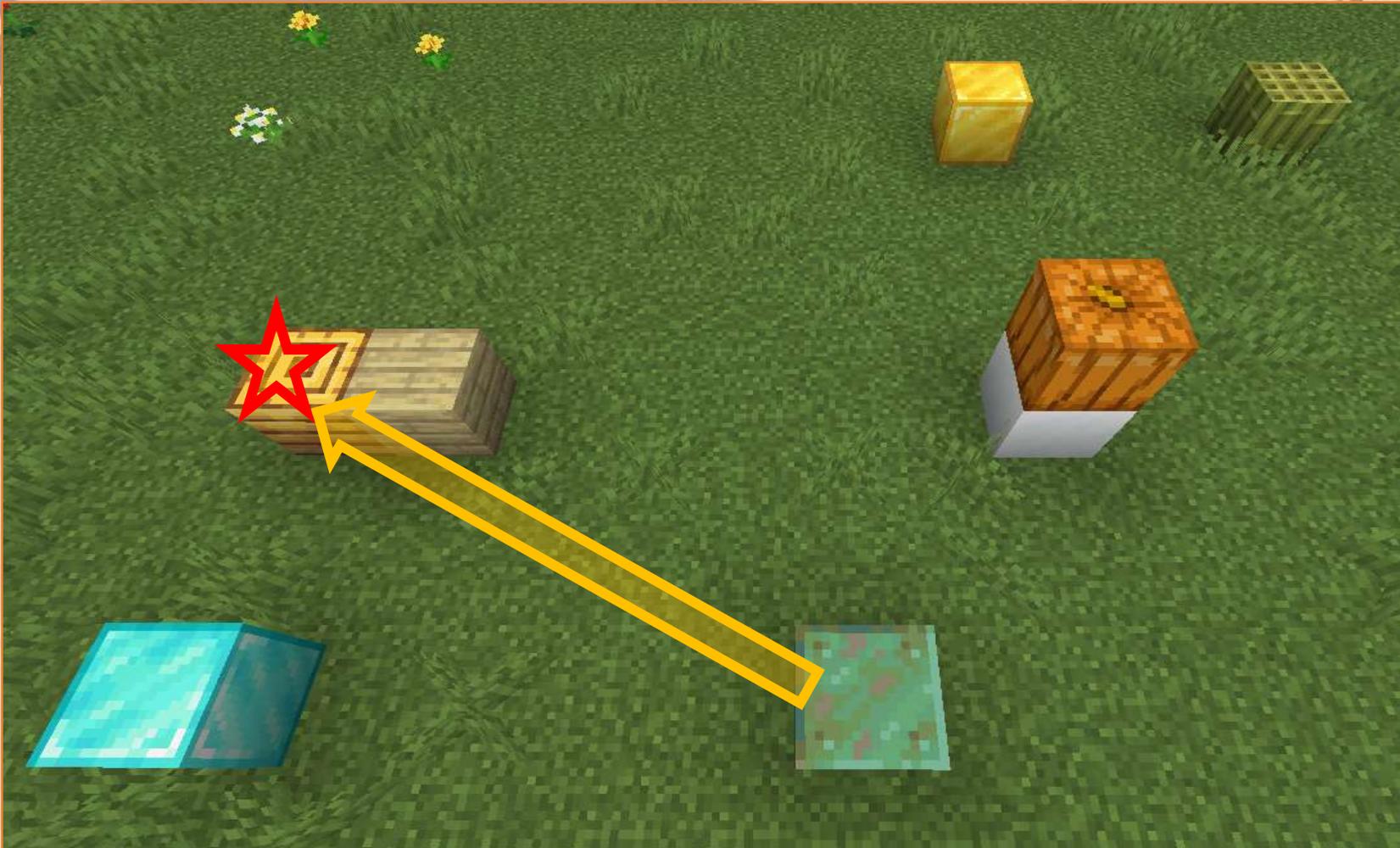
We move right



```
comando /vm  
crea un blocco fatto di Blocco di neve  
vai 5 blocchi in avanti ↑  
crea un blocco fatto di Blocco d'oro  
vai 4 blocchi a destra →  
crea un blocco fatto di Blocco di bambù  
vai alla partenza  
vai 7 blocchi a sinistra ←  
crea un blocco fatto di Alveare  
marca questo blocco  
vai 3 blocchi in dietro ↓  
crea un blocco fatto di Blocco di diamante  
vai 5 blocchi a destra →  
crea un blocco fatto di Blocco di rame  
vai alla posizione marcata  
vai 1 blocchi a destra →  
crea un blocco fatto di Assi di betulla  
vai alla partenza  
vai 1 blocchi in su ↑  
crea un blocco fatto di Lanterna di zucca
```

Marking Positions

We are going back to the marked position



```
comando /vm  
crea un blocco fatto di Blocco di neve  
vai 5 blocchi in avanti ↑  
crea un blocco fatto di Blocco d'oro  
vai 4 blocchi a destra →  
crea un blocco fatto di Blocco di bambù  
vai alla partenza  
vai 7 blocchi a sinistra ←  
crea un blocco fatto di Alveare  
marca questo blocco  
vai 3 blocchi in dietro ↓  
crea un blocco fatto di Blocco di diamante  
vai 5 blocchi a destra →  
crea un blocco fatto di Blocco di rame  
vai alla posizione marcata  
vai 1 blocchi a destra →  
crea un blocco fatto di Assi di betulla  
vai alla partenza  
vai 1 blocchi in su ↑  
crea un blocco fatto di Lanterna di zucca
```

Marking Positions

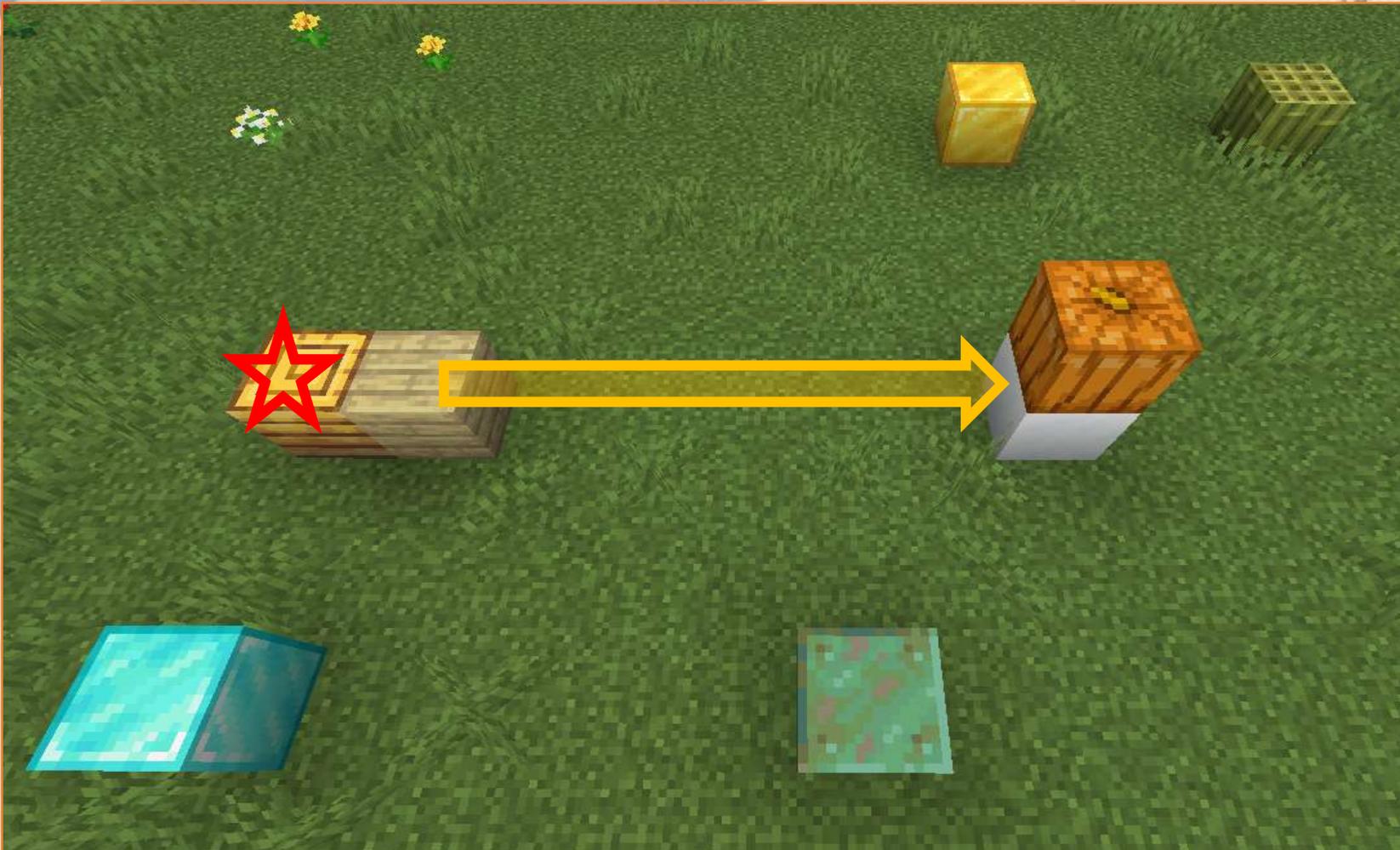
One step right ..



```
comando /vm  
crea un blocco fatto di Blocco di neve  
vai 5 blocchi in avanti ↑  
crea un blocco fatto di Blocco d'oro  
vai 4 blocchi a destra →  
crea un blocco fatto di Blocco di bambù  
vai alla partenza  
vai 7 blocchi a sinistra ←  
crea un blocco fatto di Alveare  
marca questo blocco  
vai 3 blocchi in dietro ↓  
crea un blocco fatto di Blocco di diamante  
vai 5 blocchi a destra →  
crea un blocco fatto di Blocco di rame  
vai alla posizione marcata  
vai 1 blocchi a destra →  
crea un blocco fatto di Assi di betulla  
vai alla partenza  
vai 1 blocchi in su ↑  
crea un blocco fatto di Lanterna di zucca
```

Marking Positions

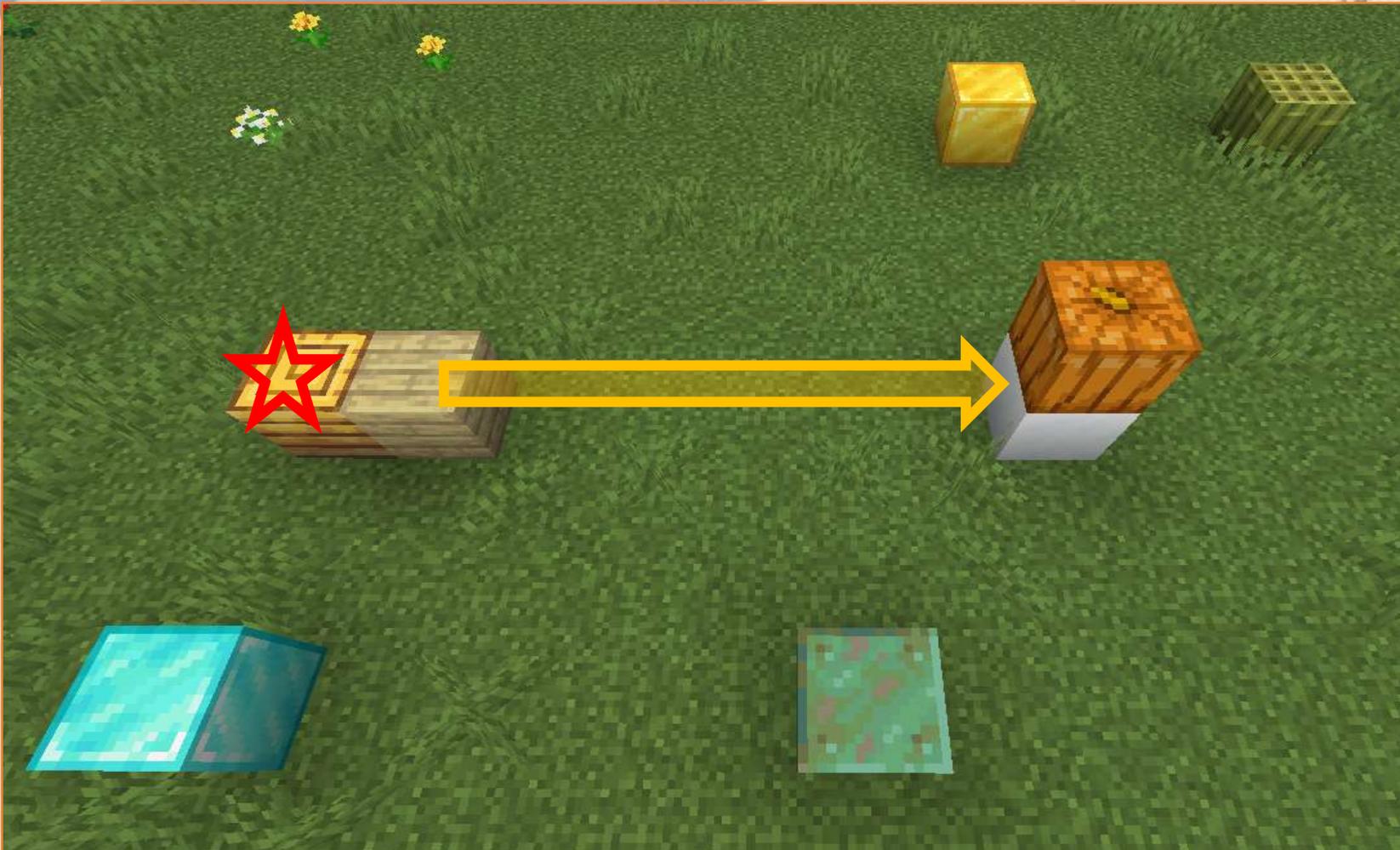
Back to the beginning



```
comando /vm  
crea un blocco fatto di Blocco di neve  
vai 5 blocchi in avanti ↑  
crea un blocco fatto di Blocco d'oro  
vai 4 blocchi a destra →  
crea un blocco fatto di Blocco di bambù  
vai alla partenza  
vai 7 blocchi a sinistra ←  
crea un blocco fatto di Alveare  
marca questo blocco  
vai 3 blocchi in dietro ↓  
crea un blocco fatto di Blocco di diamante  
vai 5 blocchi a destra →  
crea un blocco fatto di Blocco di rame  
vai alla posizione marcata  
vai 1 blocchi a destra →  
crea un blocco fatto di Assi di betulla  
vai alla partenza  
vai 1 blocchi in su ↑  
crea un blocco fatto di Lanterna di zucca
```

Marking Positions

A final pumpkin block



```
comando /vm  
crea un blocco fatto di Blocco di neve  
vai 5 blocchi in avanti ↑  
crea un blocco fatto di Blocco d'oro  
vai 4 blocchi a destra →  
crea un blocco fatto di Blocco di bambù  
vai alla partenza  
vai 7 blocchi a sinistra ←  
crea un blocco fatto di Alveare  
marca questo blocco  
vai 3 blocchi in dietro ↓  
crea un blocco fatto di Blocco di diamante  
vai 5 blocchi a destra →  
crea un blocco fatto di Blocco di rame  
vai alla posizione marcata  
vai 1 blocchi a destra →  
crea un blocco fatto di Assi di betulla  
vai alla partenza  
vai 1 blocchi in su ↑  
crea un blocco fatto di Lanterna di zucca
```

Marking Positions

Follow this example to understand the concept and importance of marking positions.



```
comando /vm  
crea un blocco fatto di Blocco di neve  
vai 5 blocchi in avanti ↑  
crea un blocco fatto di Blocco d'oro  
vai 4 blocchi a destra →  
crea un blocco fatto di Blocco di bambù  
vai alla partenza  
vai 7 blocchi a sinistra ←  
crea un blocco fatto di Alveare  
marca questo blocco  
vai 3 blocchi in dietro ↓  
crea un blocco fatto di Blocco di diamante  
vai 5 blocchi a destra →  
crea un blocco fatto di Blocco di rame  
vai alla posizione marcata  
vai 1 blocchi a destra →  
crea un blocco fatto di Assi di betulla  
vai alla partenza  
vai 1 blocchi in su ↑  
crea un blocco fatto di Lanterna di zucca
```

Follow an example

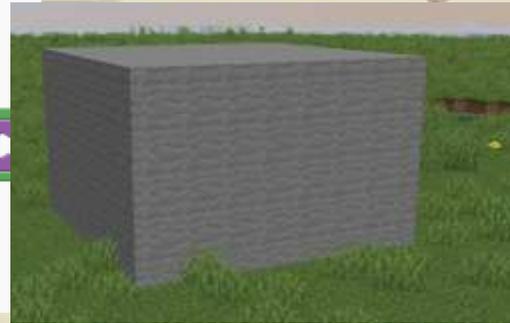
See how the robot uses marked positions to navigate and build to make a simple rabbit.



Follow an example

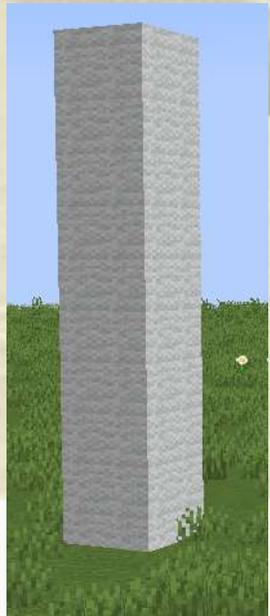
See how the robot uses marked positions to navigate and build to make a simple rabbit.

```
repeat 5 times
do
  create a full square of width 8 made of Light Grey Wool
  go 1 block up
```



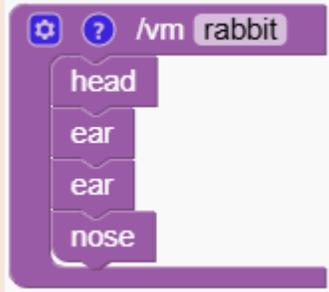
```
repeat 3 times
do
  create a full square of width 4 made of Grey Wool
  go 1 block up
```

```
repeat 10 times
do
  create a full square of width 2 made of White Wool
  go 1 block up
```



How do I put the rabbit together?

Try to recreate the rabbit as in the picture. Mark a position on top of the head and use it as a central position



```
Scratch script for 'head':
repeat 5 times
do
  create a full square of width 8 made of Light Grey Wool
  go 1 block up 1
```

```
Scratch script for 'ear':
repeat 10 times
do
  create a full square of width 2 made of White Wool
  go 1 block up 1
```

```
Scratch script for 'nose':
repeat 3 times
do
  create a full square of width 4 made of Grey Wool
  go 1 block up 1
```

How do I put the rabbit together?

This is our solution.

```

/mv rabbit
go 10 block forward ↑
head
mark this block
go 2 block left ←
ear
go to the marked block
go 2 block right →
ear
go to the marked block
go 3 block backwards ↓
go 2 block down ↓
nose

```



```

/mv head
repeat 5 times
do
create a full square of width 8 made of Light Grey Wool
go 1 block up ↑

/mv ear
repeat 10 times
do
create a full square of width 2 made of White Wool
go 1 block up ↑

/mv nose
repeat 3 times
do
create a full square of width 4 made of Grey Wool
go 1 block up ↑

```

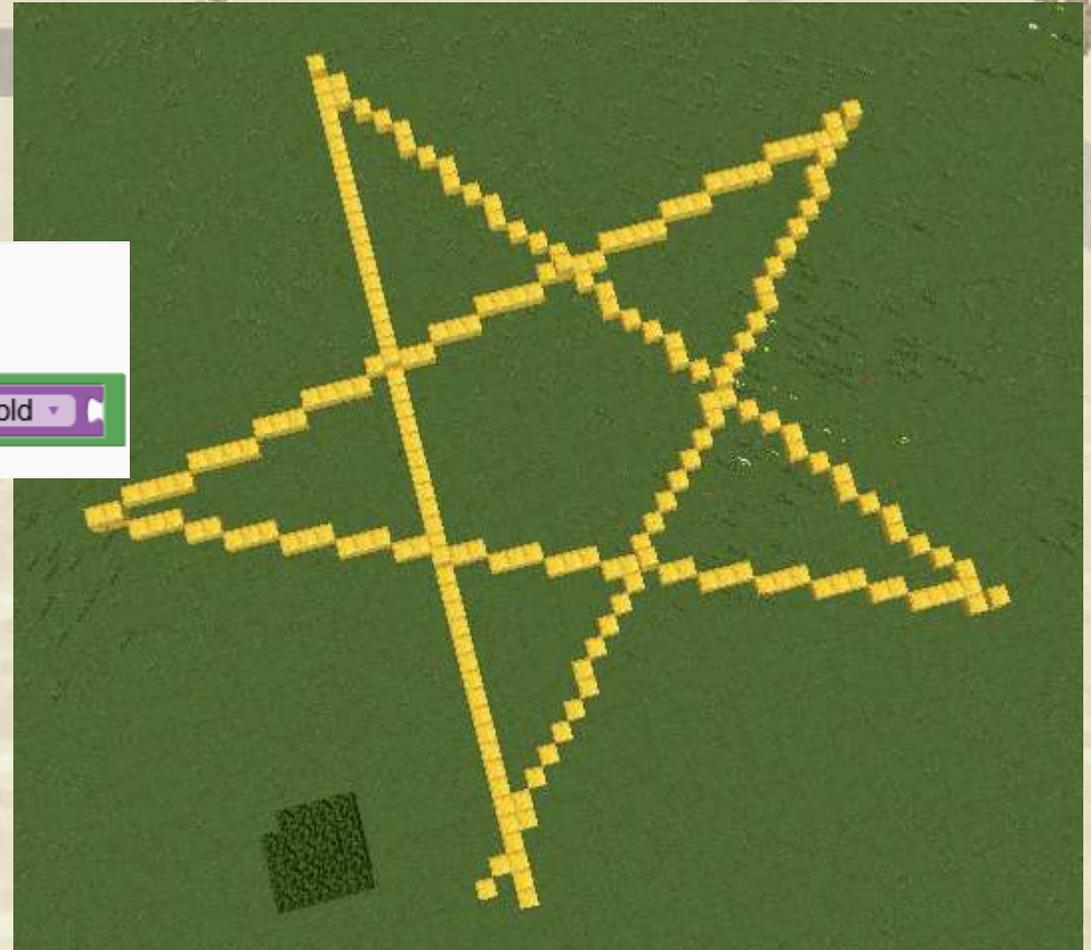
Using markings to make a star

Use connections to make a path

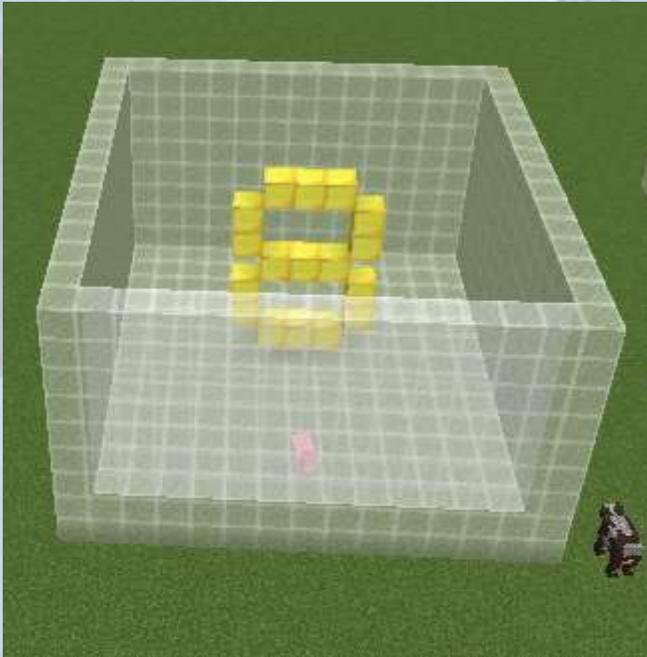
```
/vm conn
go 50 blocks forward
create a connection o--o made of Block of Gold
```

```
/vm conn
go 50 blocks forward
create a connection o--o made of Block of Gold
```

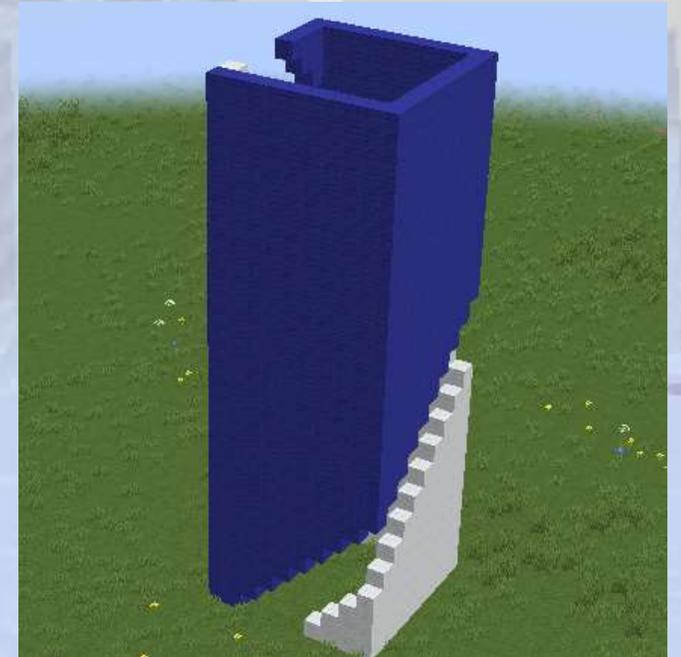
```
/vm star
repeat 5 times
do
  go 50 blocks forward
  create a connection o--o made of on the ground Block of Gold
  mark this block
  turn right by 145 degrees
```



Variables



Understand what variables are and why we need them



Variables

 Section Overview

 Objectives

We discover the concept of variables, how they are used, and their applications in coding.

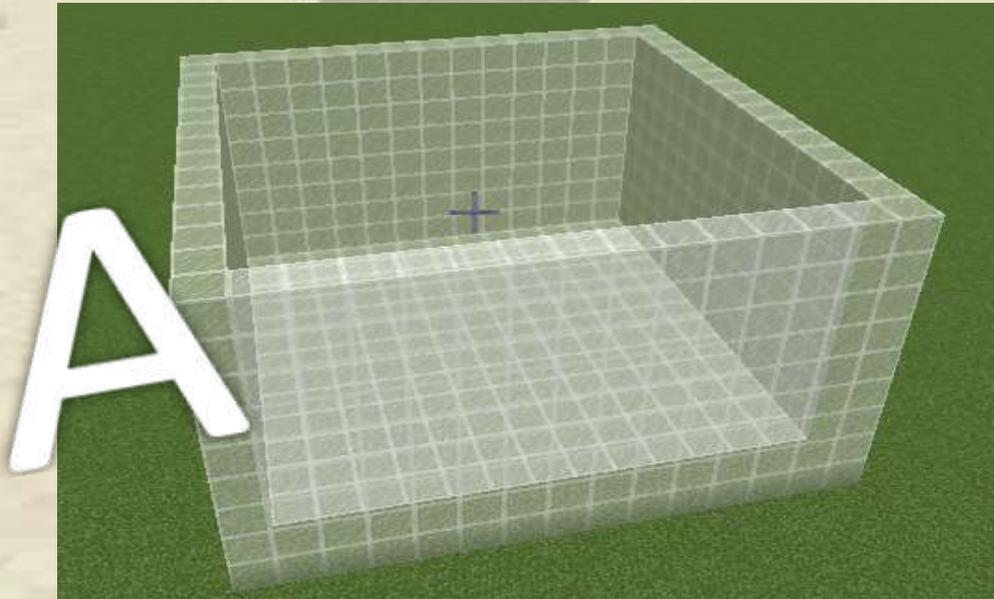
 Expected Outcomes

Students will have an understanding of what variables are and how they help in coding.

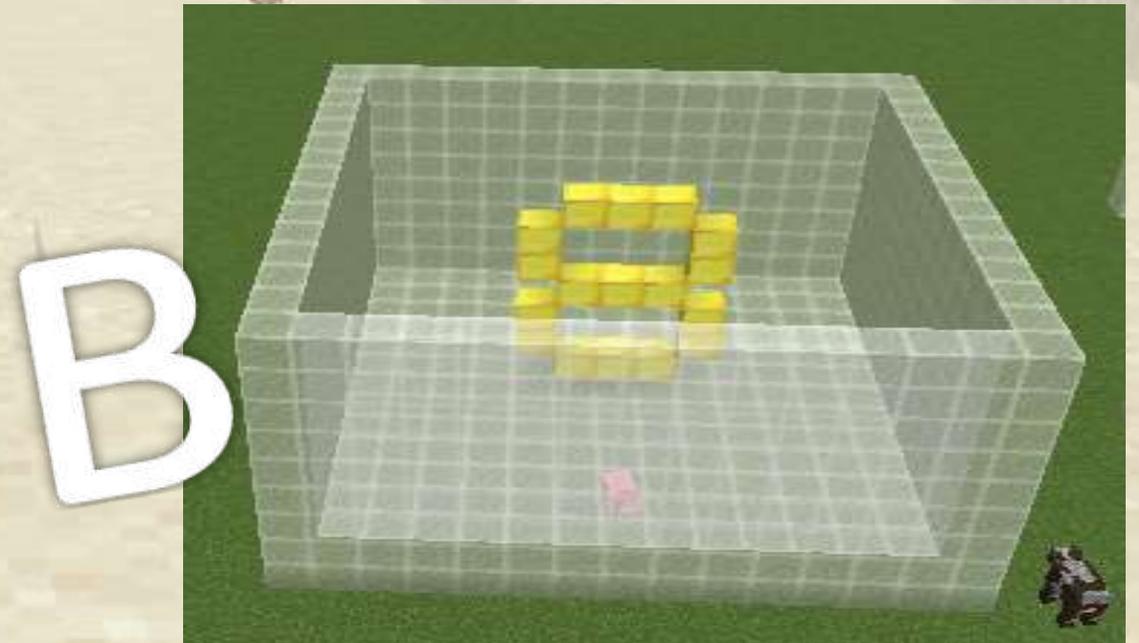
What Is a Variable?

For the computer a variable is like a box or a chest in Minecraft.

- A variable can contain only one thing.
- The computer can have many variables, so we have to give them a name.



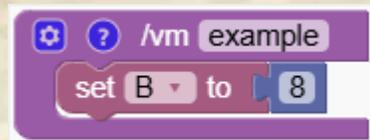
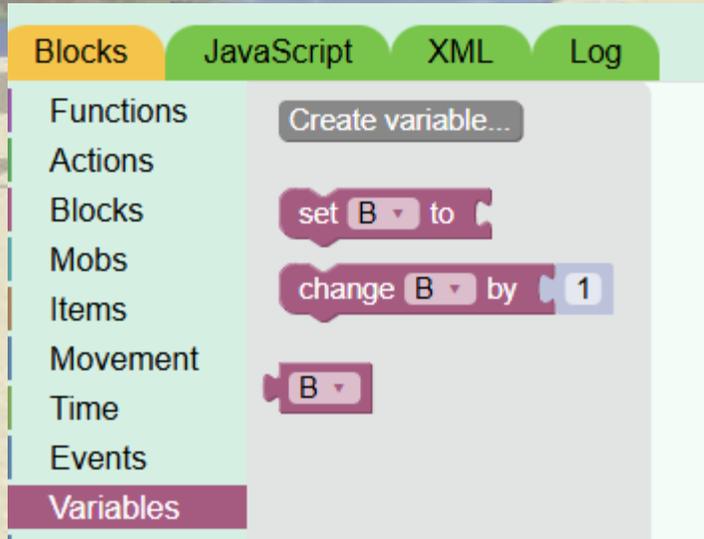
This variable is called "A" and doesn't contain anything



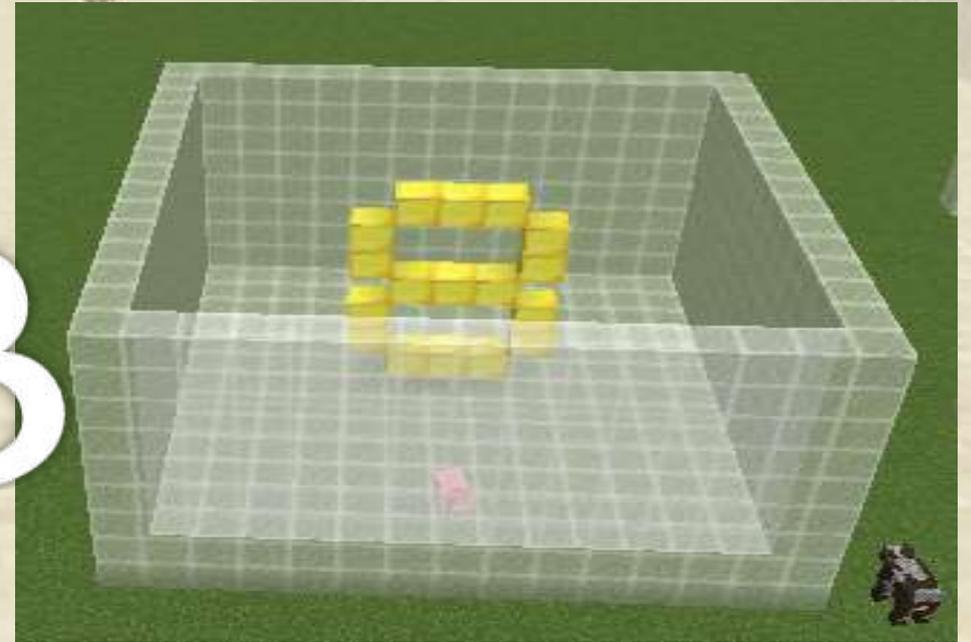
This variable is called "B" and contains the number 8

How to Create a Variable in the Editor

In the side menu under 'variables' there is the "create variable" option. Once the variable is created you'll be able to set or change it's value



B



Basic Example with a Row of Blocks

The value contained in the variable with name 'B' is the number 4

```
set B to 4  
create a row of length B made of Block of Gold
```



Now we added the number 3 to the number 4. Now B contains the number 7

```
set B to 4  
change B by 3  
create a row of length B made of Block of Gold
```



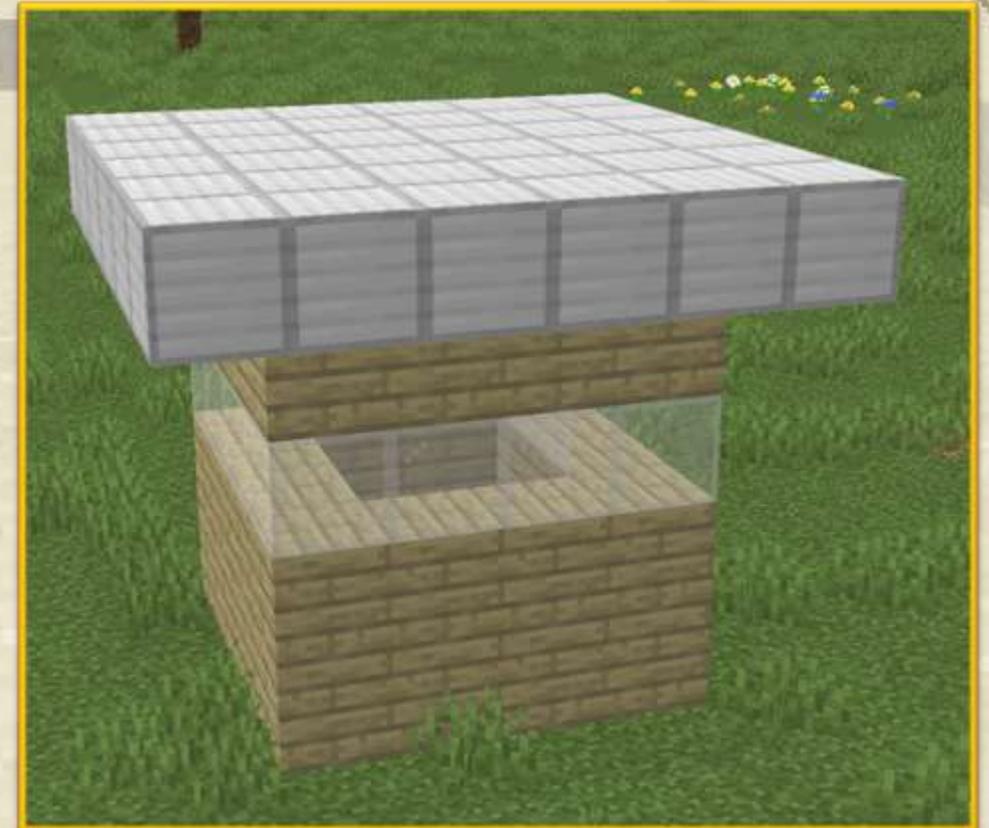
Make a House with Variable Size and Material

We are now going to see one of the advantages of using variables in defining the size and material of a house. The following program makes a simple house

```

/vm house
go 10 blocks forward
create a empty square of width 4 made of Birch Planks
go 1 blocks up
create a empty square of width 4 made of Birch Planks
go 1 blocks up
create a empty square of width 4 made of White Stained Glass
go 1 blocks up
create a empty square of width 4 made of Birch Planks
go 1 blocks up
create a empty square of width 4 made of Block of Iron

```



Make a House with Variable Size and Material

If I decide later that the house should be 8 blocks wide, I have to change the values everywhere.
What if the later I change my mind again?

```

/mv house
go 10 blocks forward
create a empty square of width 8 made of Birch Planks
go 1 blocks up
create a empty square of width 8 made of Birch Planks
go 1 blocks up
create a empty square of width 8 made of White Stained Glass
go 1 blocks up
create a empty square of width 8 made of Birch Planks
go 1 blocks up
create a empty square of width 10 made of Block of Iron

```



Make a House with Variable Size and Material

By rewriting the program using the variable "side" I can easily update the program

```

/mv house
go 10 blocks forward
set side to 8
create a empty square of width side made of Birch Planks
go 1 blocks up
create a empty square of width side made of Birch Planks
go 1 blocks up
create a empty square of width side made of White Stained Glass
go 1 blocks up
create a empty square of width side made of Birch Planks
go 1 blocks up
change side by 2
create a empty square of width side made of Block of Iron

```



Make a House with Variable Size and Material

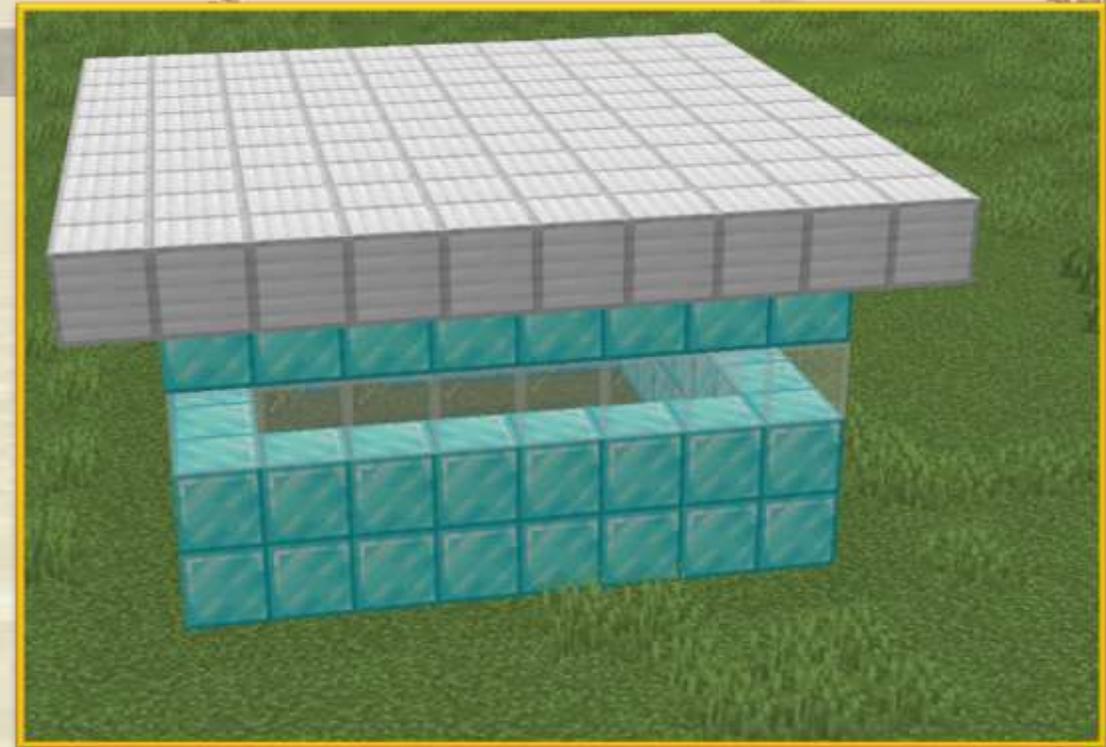
I can do the same with the blocks for the walls.

I just created a variable “mat” and put it into the program

```

? /vm house
go 10 blocks forward
set side to 8
set mat to Block of Diamond
create a empty square of width side made of mat
go 1 blocks up
create a empty square of width side made of mat
go 1 blocks up
create a empty square of width side made of White Stained Glass
go 1 blocks up
create a empty square of width side made of mat
go 1 blocks up
change side by 2
create a empty square of width side made of Block of Iron

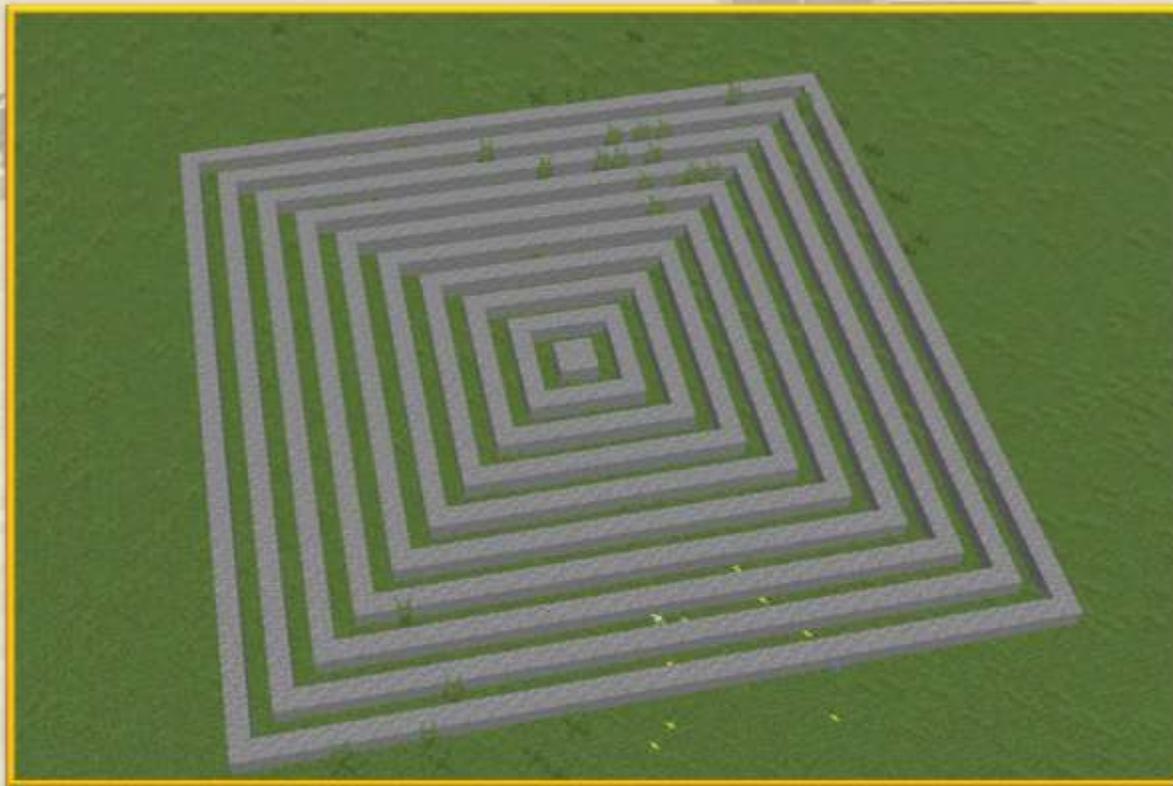
```



⚡ Make Concentric Squares

Use variables to avoid repetitive tasks.

We want to create the following shape. How shall we do it?



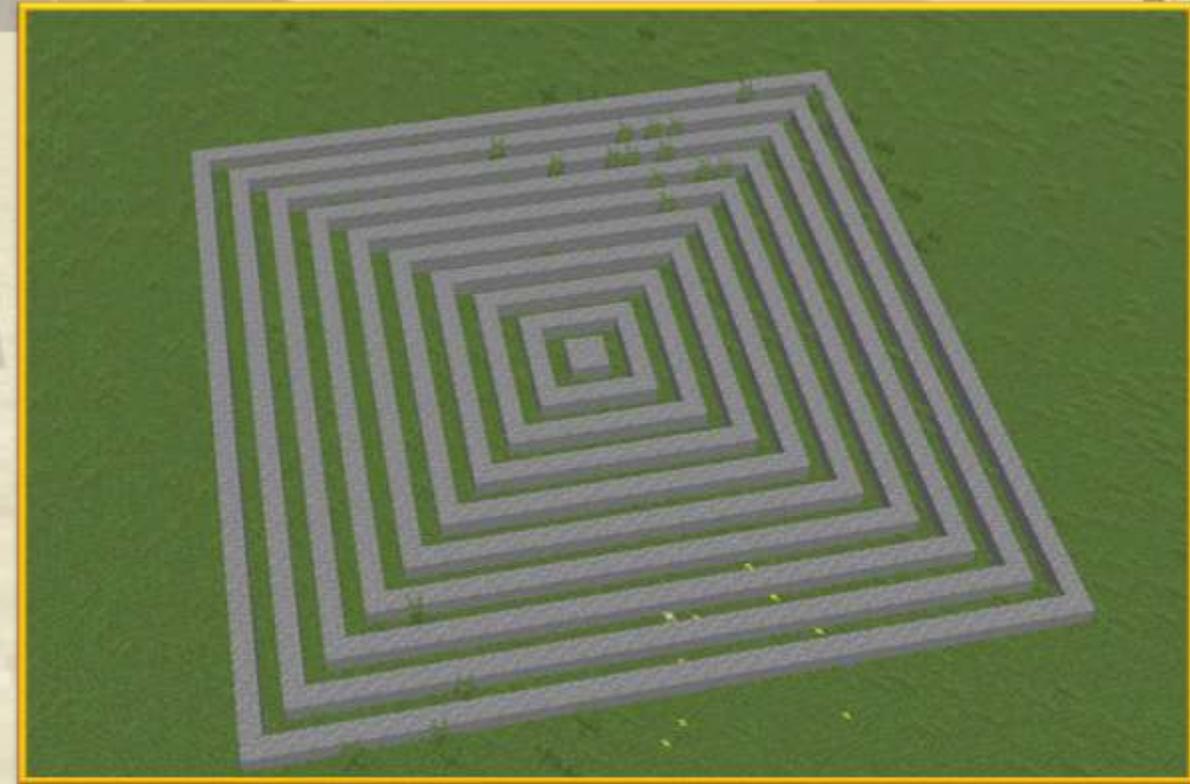
⚡ Make Concentric Squares

This is a slow, repetitive and poor solution

```

? /vm quad
create a empty square of width 2 made of Andesite
create a empty square of width 6 made of Andesite
create a empty square of width 8 made of Andesite
create a empty square of width 12 made of Andesite
create a empty square of width 16 made of Andesite
create a empty square of width 20 made of Andesite

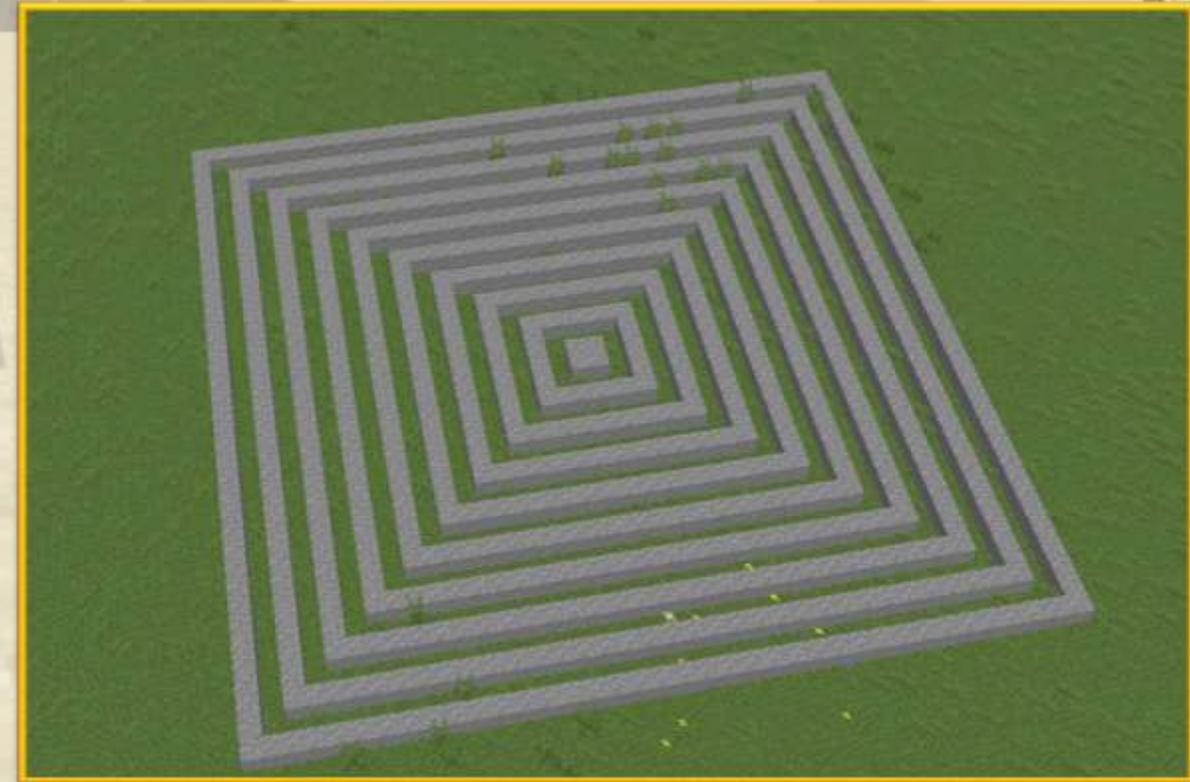
```



⚡ Make Concentric Squares

Now we use a variable but this doesn't help. The program is still too long.

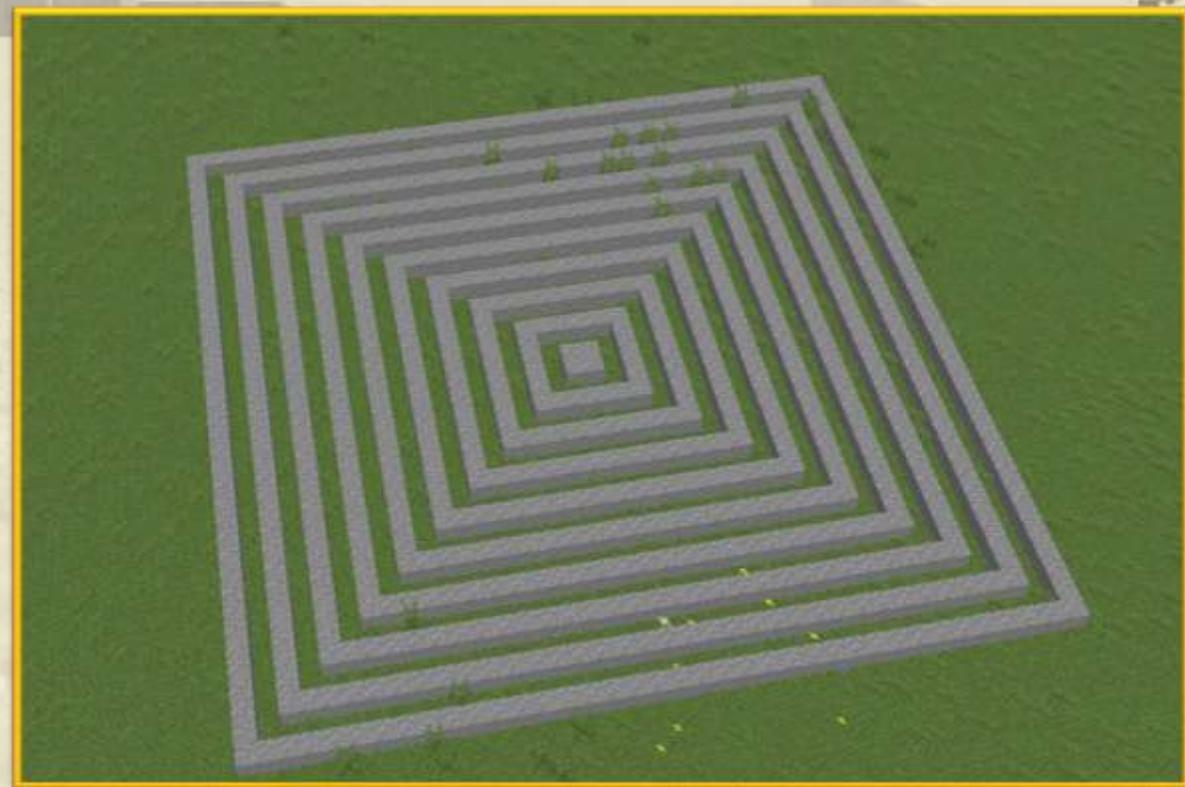
```
set side to 2
create a empty square of width side made of Andesite
change side by 4
create a empty square of width side made of Andesite
change side by 4
create a empty square of width side made of Andesite
change side by 4
create a empty square of width side made of Andesite
change side by 4
create a empty square of width side made of Andesite
change side by 4
create a empty square of width side made of Andesite
```



⚡ Make Concentric Squares

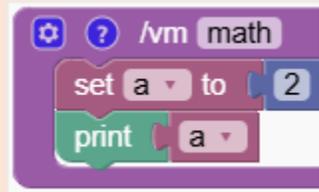
This is a much better program.

```
vm quad
set side to 2
repeat 10 times
do
  create a empty square of width side made of Andesite
  change side by 4
```

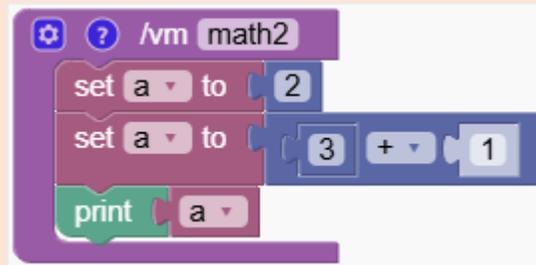


What Numbers Are Generated by This Code?

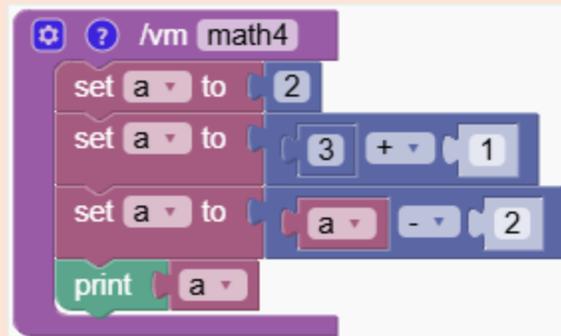
Practice modifying values in variables.



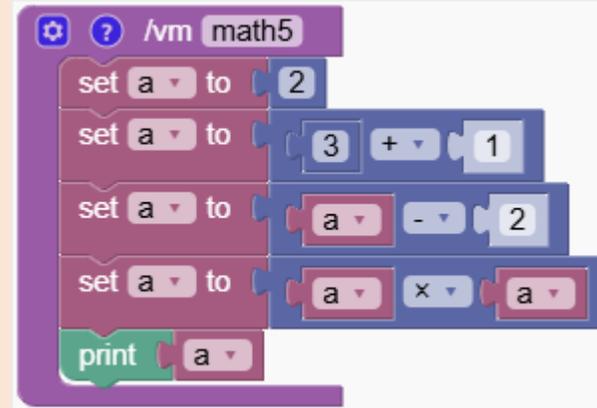
```
set a to 2
print a
```



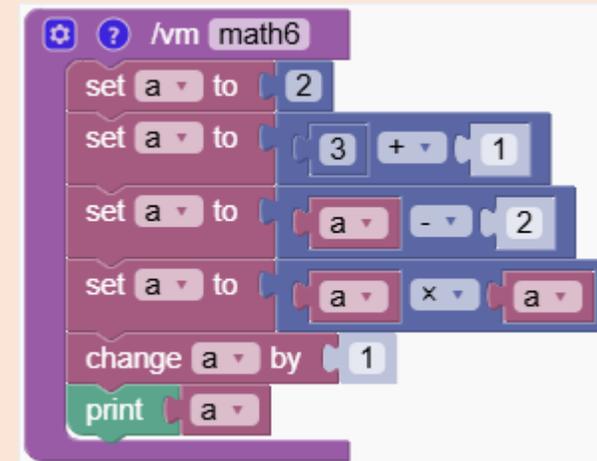
```
set a to 2
set a to 3 + 1
print a
```



```
set a to 2
set a to 3 + 1
set a to a - 2
print a
```



```
set a to 2
set a to 3 + 1
set a to a - 2
set a to a x a
print a
```

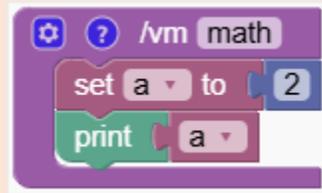


```
set a to 2
set a to 3 + 1
set a to a - 2
set a to a x a
change a by 1
print a
```

Quiz

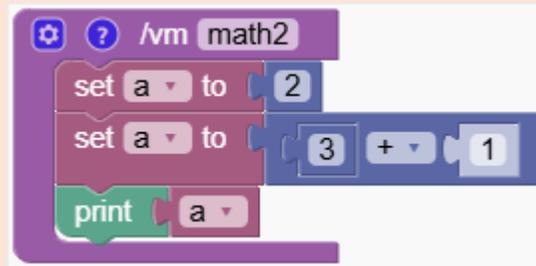
What Numbers Are Generated by This Code?

Practice modifying values in variables.



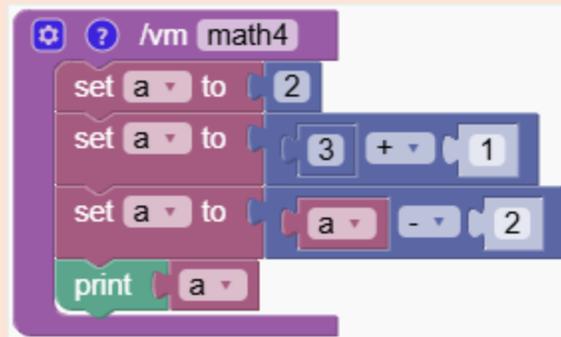
```
set a to 2
print a
```

2



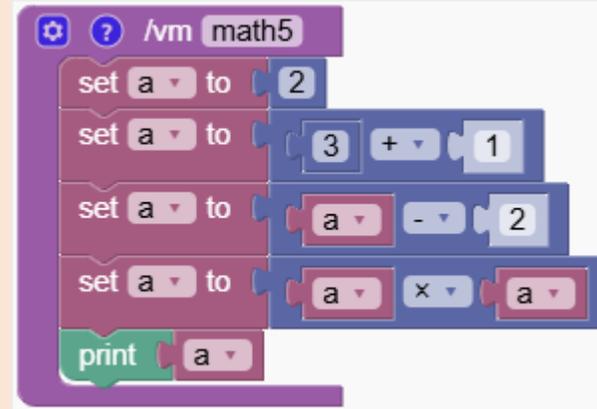
```
set a to 2
set a to 3 + 1
print a
```

4



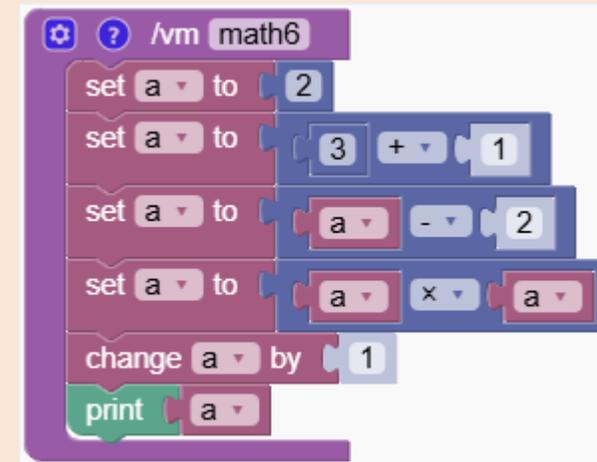
```
set a to 2
set a to 3 + 1
set a to a - 2
print a
```

2



```
set a to 2
set a to 3 + 1
set a to a - 2
set a to a x a
print a
```

4



```
set a to 2
set a to 3 + 1
set a to a - 2
set a to a x a
change a by 1
print a
```

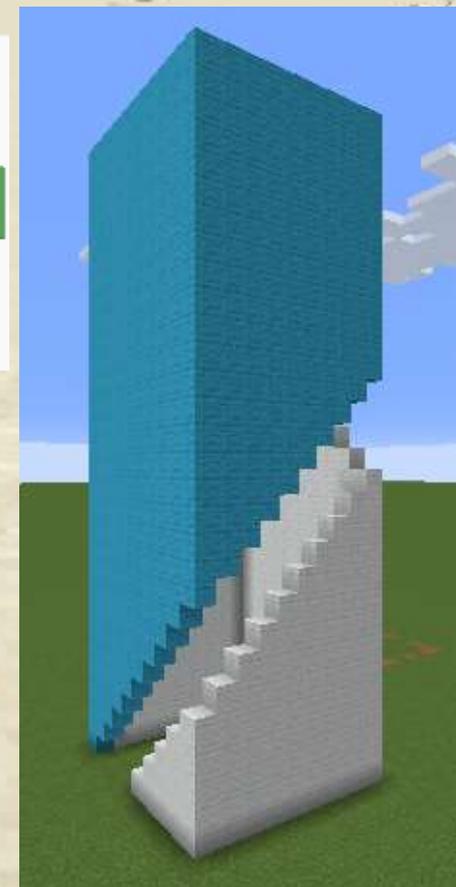
5

Quiz

⚡ Create a Parkour

Transform towers into exciting parkour challenges using variables.

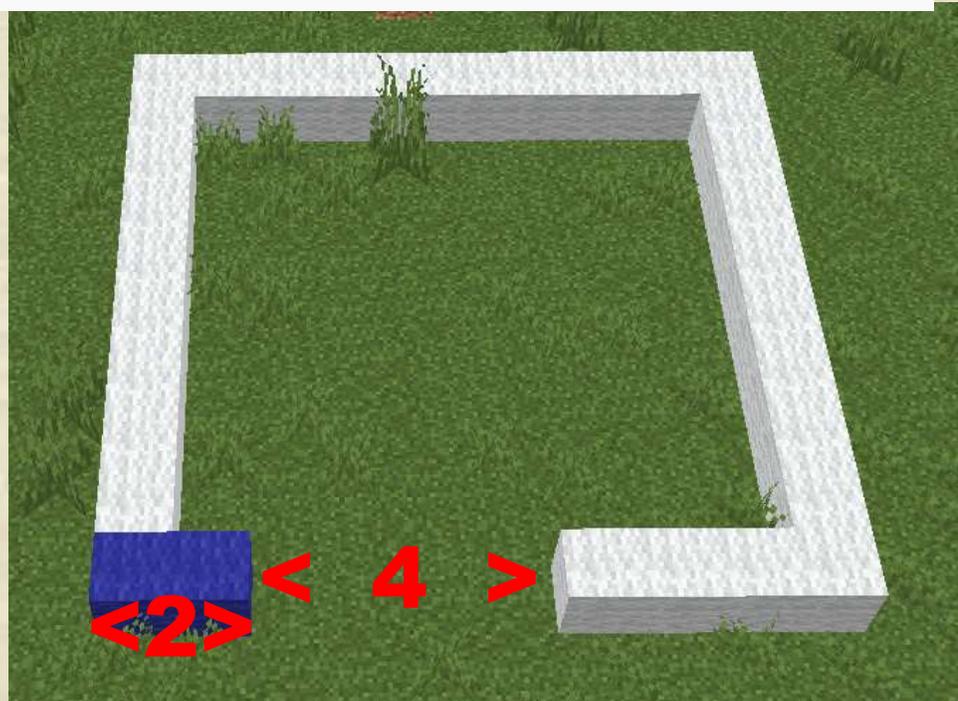
```
/vm pk
repeat 30 times
do
  create a empty square of width 10 made of 4 of Light Blue Wool 4 of Air 100 of White Wool
  go 1 blocks up
```



⚡ Create a Parkour

We start by making a square made of 2 blue blocks, followed by 4 blocks of air.
We also provide many white blocks to fill up the rest of the structure

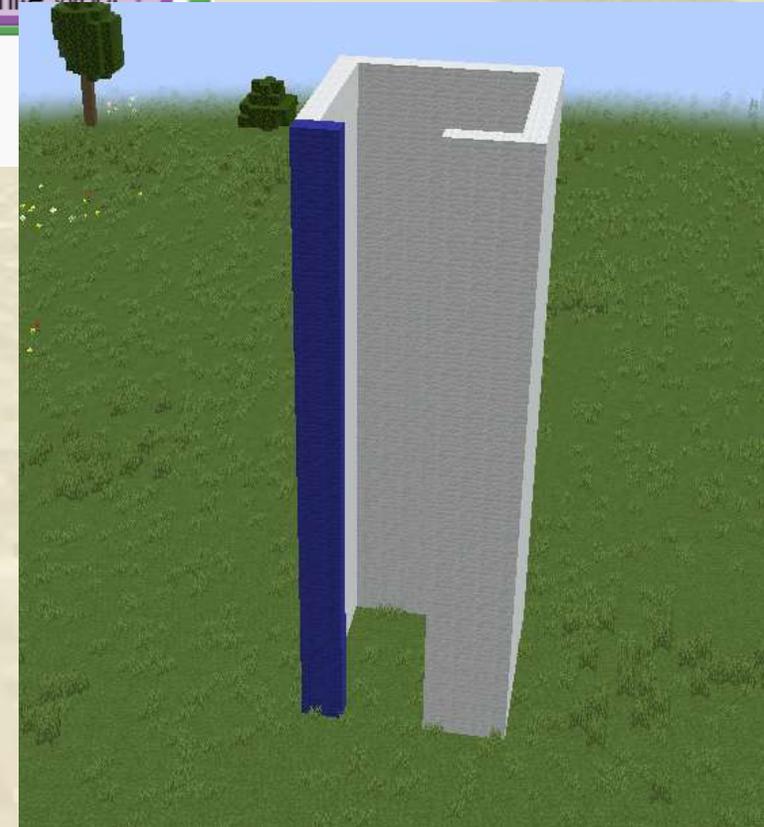
```
create a empty square of width 10 made of 2 of Blue Wool 4 of Air 40 of White Wool
```



⚡ Create a Parkour

Now we make it into a tower by repeating it 30 times

```
repeat 30 times  
do  
  create a empty square of width 10 made of 2 of Blue Wool 4 of Air 40 of White Wool  
  go 1 blocks up
```



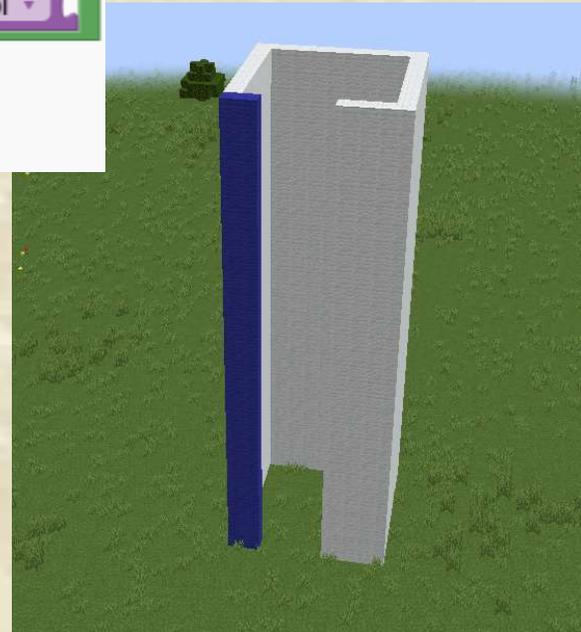
⚡ Create a Parkour

We want to add more blue blocks every time we go up one layer.
To prepare this, we replace the number “2” with a new variable called “num”.

```
set num to 2
repeat 30 times
do
  create a empty square of width 10 made of num of Blue Wool 4 of Air 40 of White Wool
  go 1 blocks up ↑
```

Put the number 2 inside the variable 'num'

Use the 'num of' block

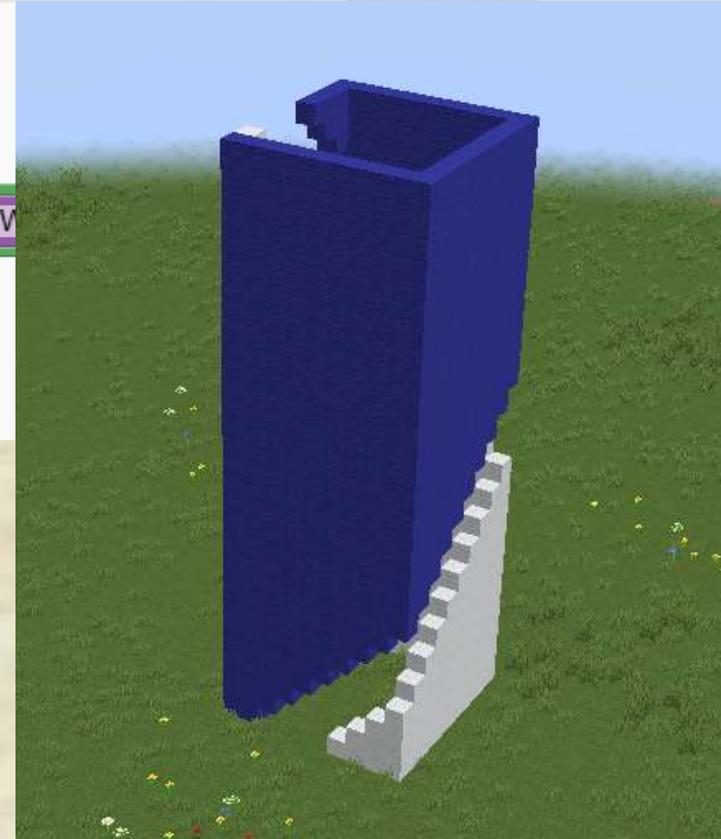


⚡ Create a Parkour

Now we change the value inside the variable 'num' so that at every new layer the number of blue blocks becomes bigger

```
set num to 2
repeat 30 times
do
  create a empty square of width 10 made of num of Blue Wool 4 of Air 40 of V
  go 1 blocks up 1
  change num by 1
```

Change the number inside 'num' at every level



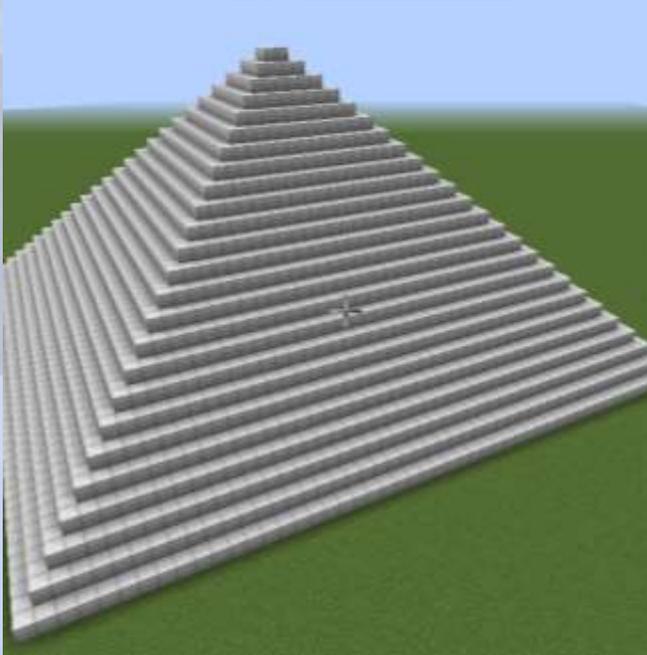
⚡ Create a Parkour

Let' add some challenge with spiders and a treasure !

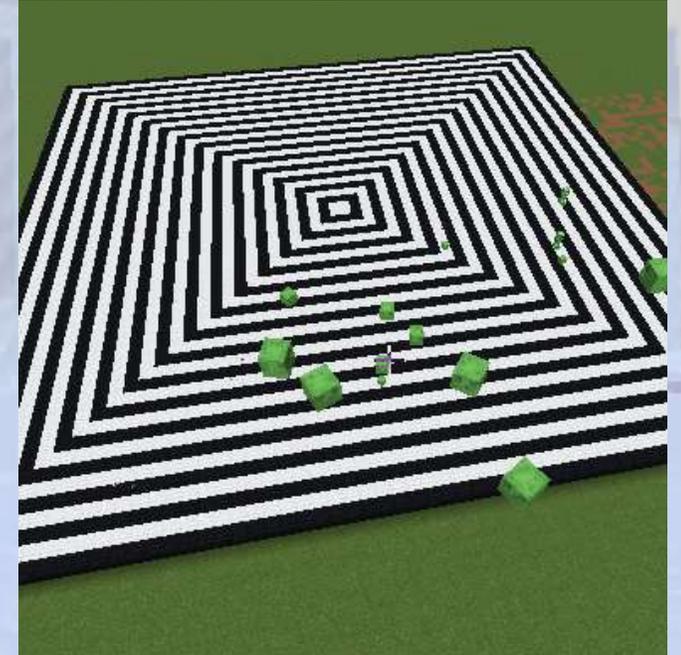
```
vm pk
set num to 2
repeat 30 times
do
  create a empty square of width 10 made of num of Blue Wool 4 of Air 40 of White Wool
  go 1 blocks up 1
  change num by 1
  create a row of length 6 made of White Wool
  go 1 blocks up 1
  create a row of length 5 made of Spider
  create chest with Golden Apple
```



Counting Loops



Learn to use the
“for” loop



Counting Loops

 Section Overview

 Objectives

Explore how the for loops work and how to use them effectively.

 Expected Outcomes

What Are We Going to Learn

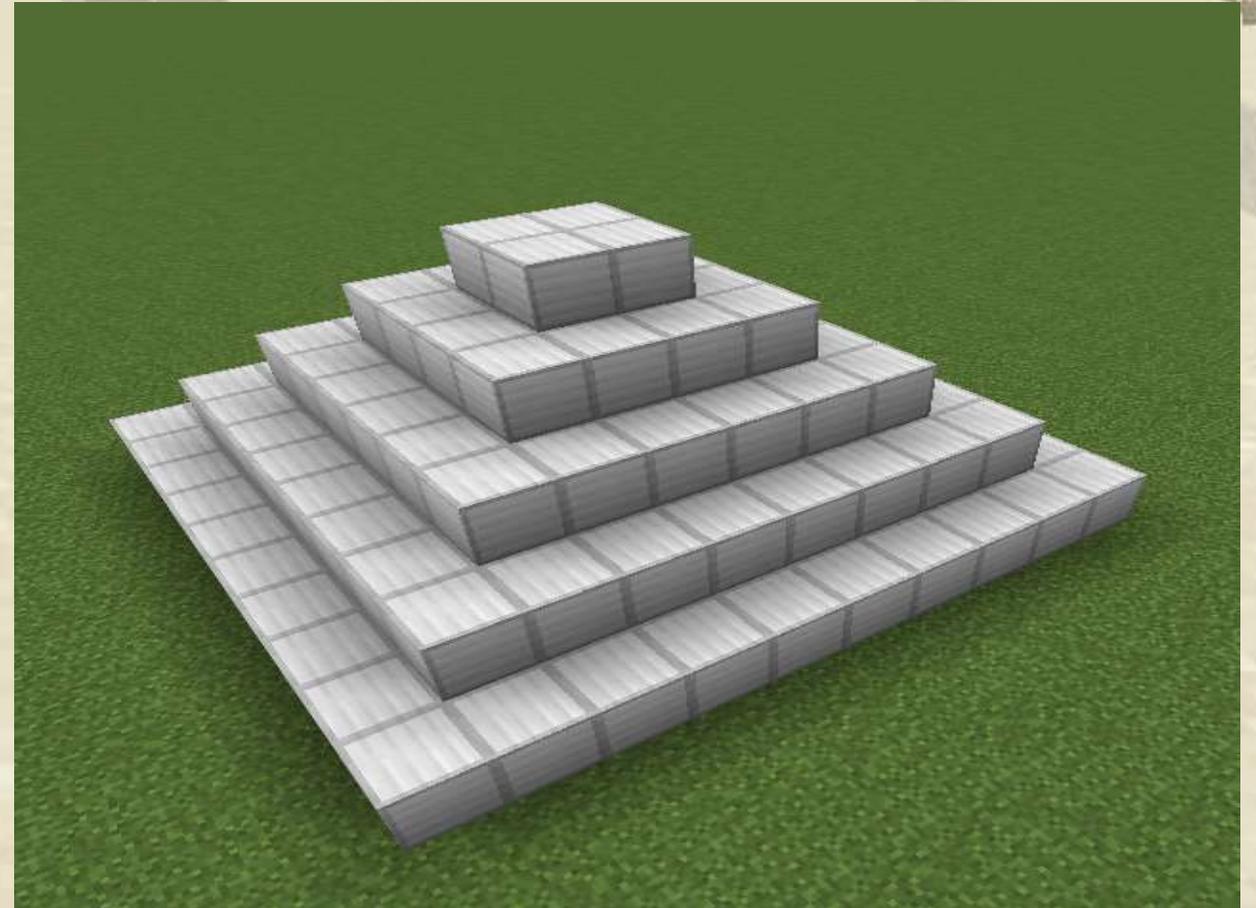
We focus on using loops to automate repetitive tasks and build efficient programs.



Coding a Pyramid

To create a pyramid we can use the following program but this is a poor solution if the pyramid should be much higher.

```
pyramid
create a empty square of width 10 made of Block of Iron
go 1 blocks up
create a empty square of width 8 made of Block of Iron
go 1 blocks up
create a empty square of width 6 made of Block of Iron
go 1 blocks up
create a empty square of width 4 made of Block of Iron
go 1 blocks up
create a empty square of width 2 made of Block of Iron
```



Coding a Pyramid

This solution is better. We use a variable to keep track of the width of the pyramid
After creating a level, we decrease the width by 2

```
vm pyramid
set width to 50
repeat 25 times
do
  create a empty square of width 2 made of Block of Iron
  go 1 blocks up 1
  change width by -2
```



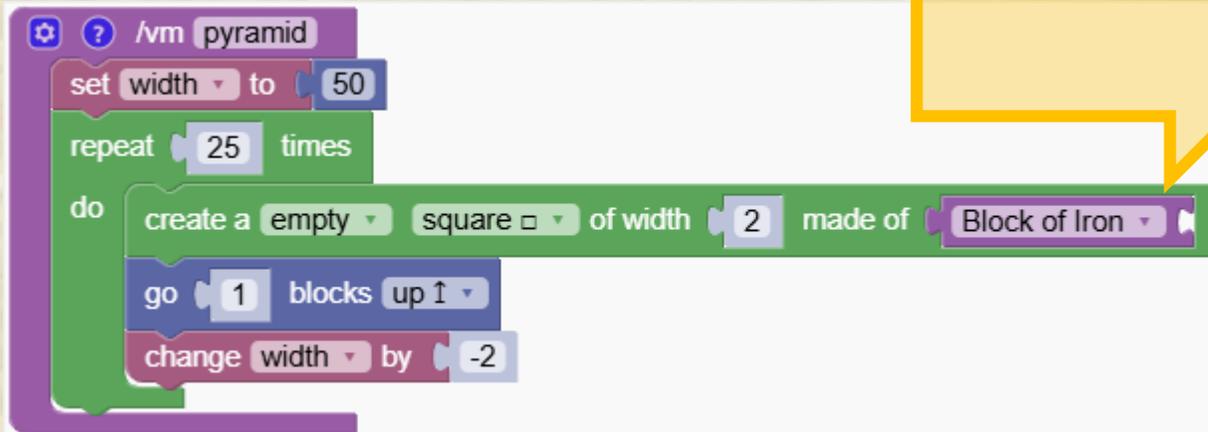
The counting loop

To create the pyramid we used the program on the left.

This type of program is very common in coding and therefore it exists a more advanced loop to support it.

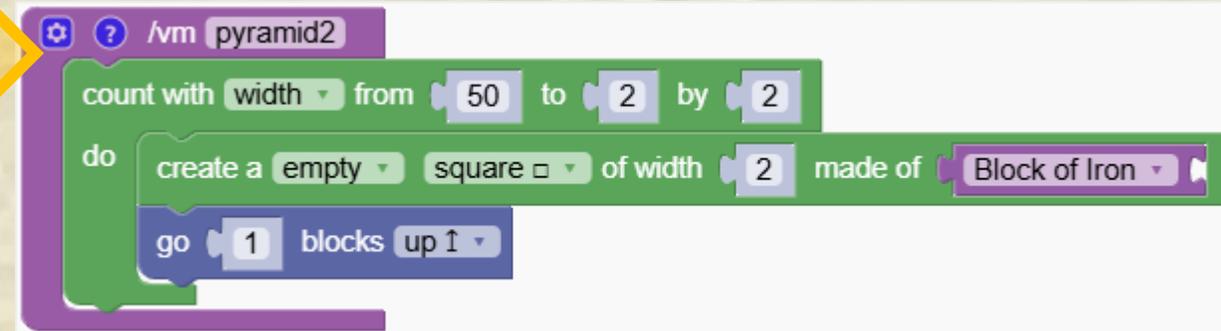
The counting loop.

The program of the right does the same job of creating the pyramid by using the counting loop.



```
set width to 50
repeat 25 times
do
  create a empty square of width 2 made of Block of Iron
  go 1 blocks up 1
  change width by -2
```

The code for 'pyramid' uses a 'repeat' loop. It starts by setting 'width' to 50. Then it repeats 25 times: create a 2x2 empty square of 'Block of Iron', go up 1 block, and decrease 'width' by 2.



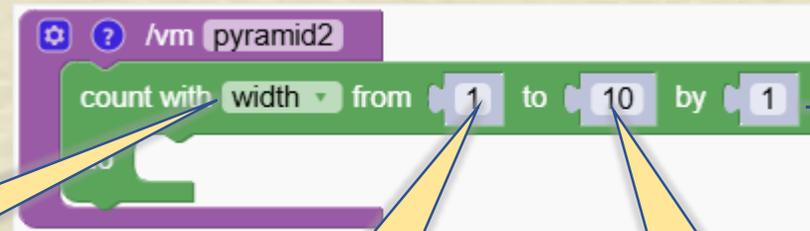
```
count with width from 50 to 2 by 2
do
  create a empty square of width 2 made of Block of Iron
  go 1 blocks up 1
```

The code for 'pyramid2' uses a 'count with' loop. It counts 'width' from 50 down to 2 by 2. Inside the loop, it creates a 2x2 empty square of 'Block of Iron' and goes up 1 block.

The counting loop

The counting loop is similar to our standard repeat loop but has the following values:

- Automatically creates a variable
- We can set the start and end value
- We can set the step to add when changing the variable



**Variable name
'width'**

**Start value
1**

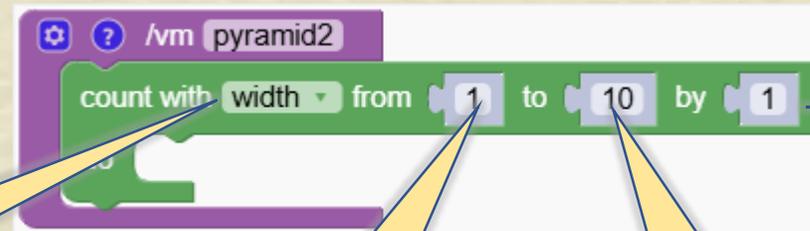
**End value
10**

**Step to add
1**

The counting loop

The counting loop is similar to our standard repeat loop but has the following values:

- Automatically creates a variable
- We can set the start and end value
- We can set the step to add when changing the variable



**Variable name
'width'**

**Start value
1**

**End value
10**

**Step to add
1**

Make Arrows

Here are some examples. Which numbers are printed when we run the programs?

```
count with i from 1 to 5 by 1
do print i
```



```
1
2
3
4
5
```

```
count with i from 20 to 30 by 2
do print i
```



```
count with i from 10 to 5 by 1
do print i
```



Make Arrows

Here are some examples. Which numbers are printed when we run the programs?

```
1 /vm count
2 count with i from 1 to 5 by 1
3 do print i
```



```
1 1
2 2
3 3
4 4
5 5
```

```
1 /vm count
2 count with i from 20 to 30 by 2
3 do print i
```



```
1 20
2 22
3 24
4 26
5 28
6 30
```

```
1 /vm count
2 count with i from 10 to 5 by 1
3 do print i
```



```
1 10
2 9
3 8
4 7
5 6
6 5
```

The counting loop

Our pyramid was created starting at width 50 at the bottom and width 2 at the top. We looped with a variable called 'loop' and at every cycle we reduced it by 2



```
count with width from 50 to 2 by 2  
create a empty square of width 2 made of Block of Iron  
go 1 blocks 1
```

Variable name
'width'

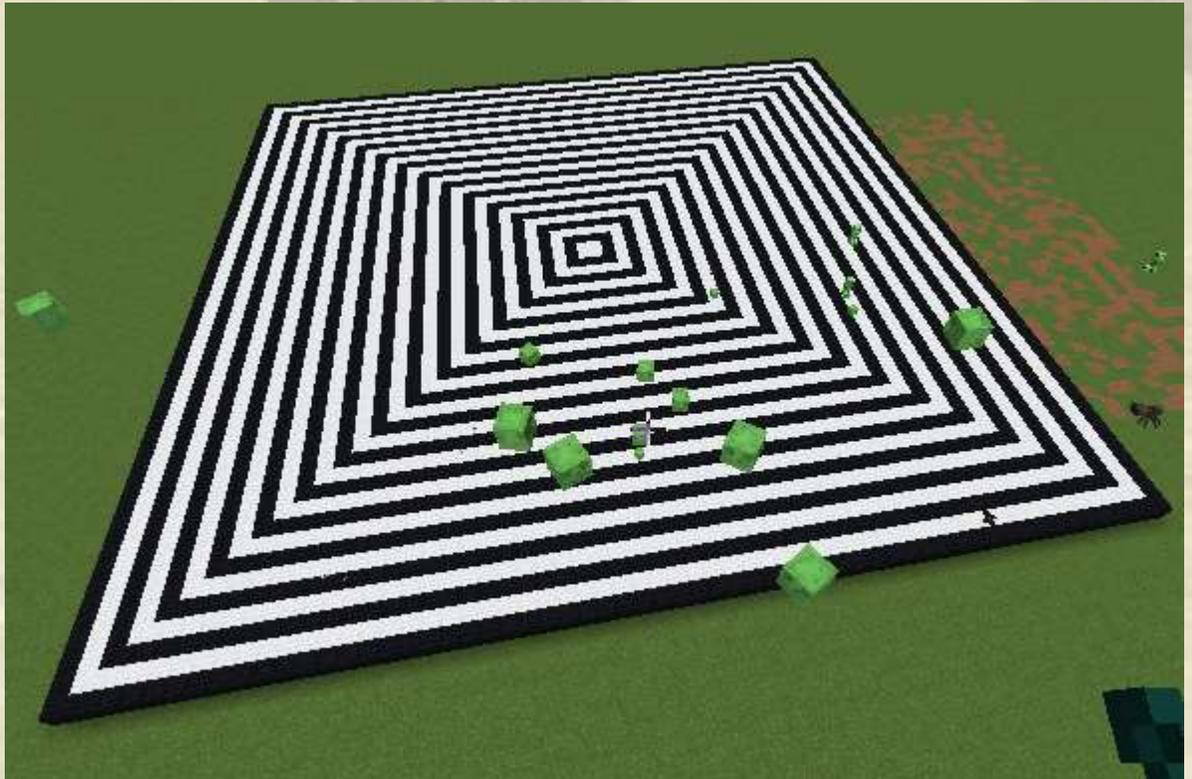
Start value
50

End value
2

Step to add
2

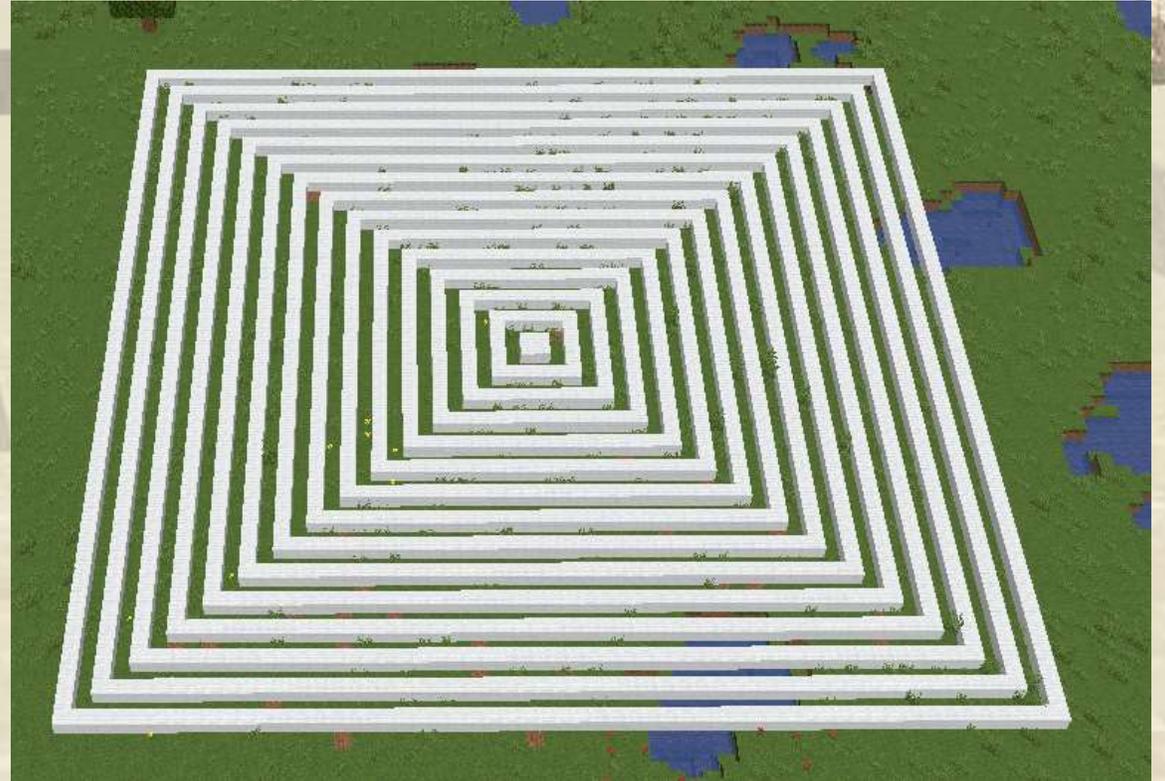
⚡ Fun Carpets

Have fun creating colorful carpets with simple loops.



Make Your Own Carpets

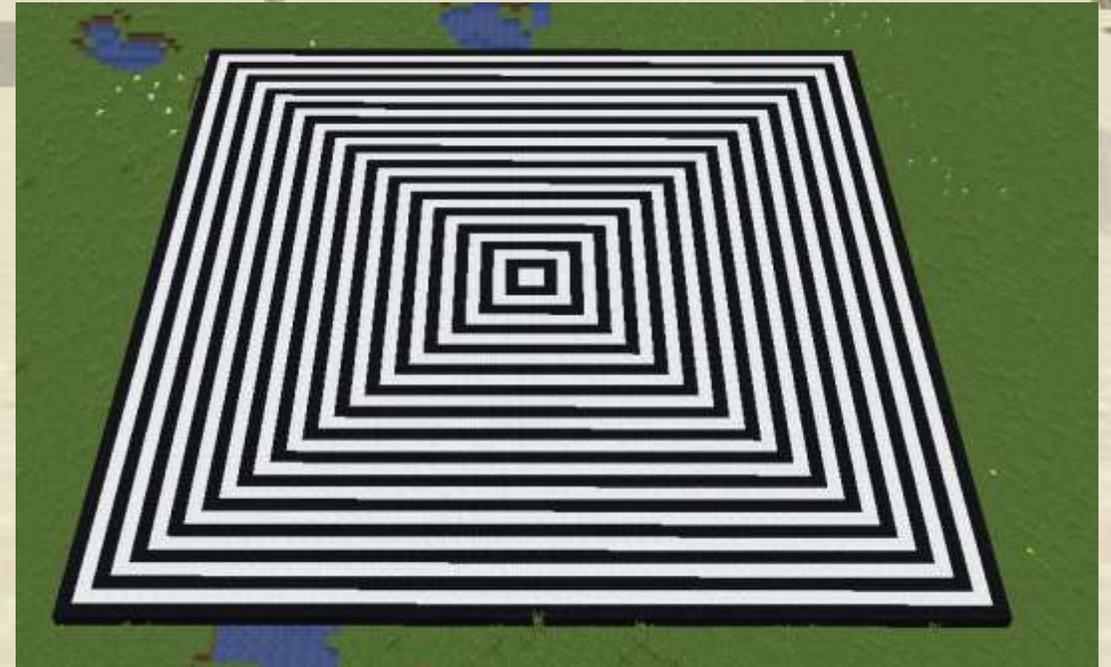
Create simple two-colored carpets using loops.



```
function /vm carpet
  count with i from 2 to 60 by 4
  do
    create a empty square of width i made of White Wool
```

Make Your Own Carpets

Create simple two-colored carpets using loops.



```

? /vm carpet
count with i from 2 to 60 by 4
do
  create a empty square of width i made of White Wool
count with i from 4 to 60 by 4
do
  create a empty square of width i made of Black Wool

```

How do I add another color?

This time you must change not only the start value but also the step of the loop .

Can you add even more colors?



Quiz

How do I add another color?

This time you must change not only the start value but also the step of the loop .

Can you add even more colors?

```

? /vm carpet
count with i from 2 to 60 by 6
do create a empty square of width i made of White Wool
count with i from 4 to 60 by 6
do create a empty square of width i made of Black Wool
count with i from 6 to 60 by 6
do create a empty square of width i made of Red Wool

```



Quiz

⚡ The Sandclock

Use counting loops to create a sandclock.



⚡ The Sandclock

First we create the bottom part.

We want the sandclock to grow wide, therefore we do smaller steps and we go up only in half steps (0.5)

```

? /vm sandclock
count with i from 15 to 2 by 0.5
do
  create a empty circle of radius i made of Block of Gold
  go 0.5 blocks up 1

```



⚡ The Sandclock

Now we repeat the program, we just swap the start and end values of the loop

```
vm sandclock
count with i from 15 to 2 by 0.5
do
  create a empty circle of radius i made of Block of Gold
  go 0.5 blocks up 1
count with i from 2 to 15 by 0.5
do
  create a empty circle of radius i made of Block of Gold
  go 0.5 blocks up 1
```

The code block shows a program named 'sandclock' with four main blocks. The first block is a 'count with i from 15 to 2 by 0.5' loop. The second block is a 'do' loop containing 'create a empty circle of radius i made of Block of Gold' and 'go 0.5 blocks up 1'. The third block is a 'count with i from 2 to 15 by 0.5' loop. The fourth block is a 'do' loop containing 'create a empty circle of radius i made of Block of Gold' and 'go 0.5 blocks up 1'. Four yellow arrows point to the '15', '2', '2', and '15' values in the loops, indicating the swapped start and end values.



Logic and Conditionals



“If .. then .. Else” and
random numbers



Random Numbers

Section Overview

In this section, we will learn how to use logic blocks to introduce conditions and control the flow of a program.

Objectives

We learn how to put rules in our code, using if-else statements and other logic to make decisions based on conditions.

Explanation of the block that generates random numbers, which can add unpredictability to our programs.

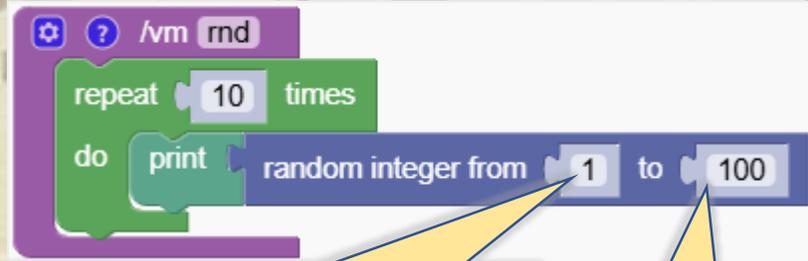
Expected Outcomes

Random Numbers

A random number is a number chosen unpredictably, like rolling a dice.

The block picks a new number each time you run the program!

You can set the minimum and maximum value of the possible numbers

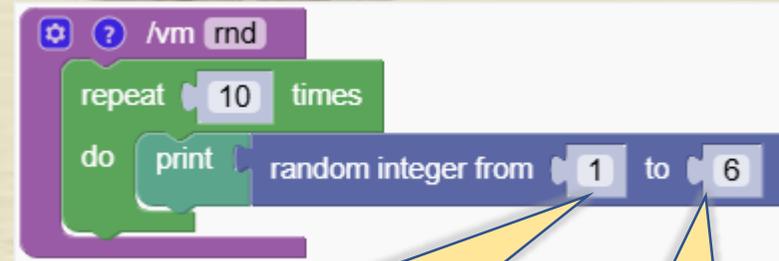


```
Scratch code block: /vm rnd, repeat 10 times, do print random integer from 1 to 100
```

Lowest value 1

Highest value 100

23
44
2
67
66
98
7
15
23
90



```
Scratch code block: /vm rnd, repeat 10 times, do print random integer from 1 to 6
```

Lowest value 1

Highest value 6

4
1
6
4
3
2
3
6
5
1

⚡ Artistic towers

This code generates artistic towers.

Make your own art by changing the values and using circles or polygons. Have fun!

```
random
repeat 20 times
do
  create a full square of width random integer from 2 to 8 made of Block of Copper
  go 1 blocks up
```



⚡ Spreading Flowers

In this example, we use the random numbers to move the robot in many different places to plant flowers.

```
random
repeat 500 times
do
  turn right by random integer from 1 to 360 degrees
  go random integer from 5 to 25 blocks forward
  create a block = made of Moon Daisy
  go to the start
```

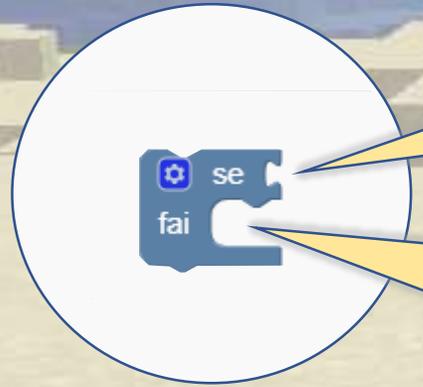
Rotate in any direction

Move forward



The Logic Blocks

Learn about the if, else, and elsif blocks, which are fundamental for making decisions in code.



Put here the condition

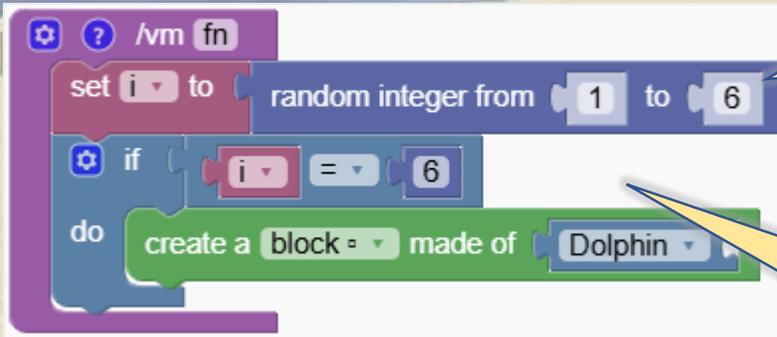
**Put here what you want to do if
the condition is true**



**Example:
If the robot is in a block made of
water then spawn a dolphin**

The Logic Blocks

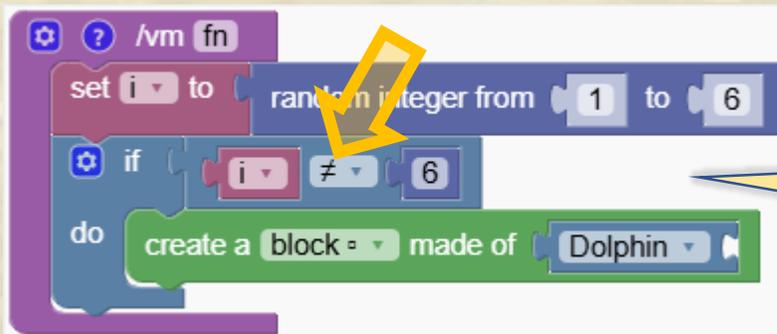
The logic blocks are used very often with variables



```
set i to random integer from 1 to 6
if i = 6
do create a block made of Dolphin
```

put in the variable "i" a value between 1 and 6

Create a dolphin only if the variable "i" contains the value 6

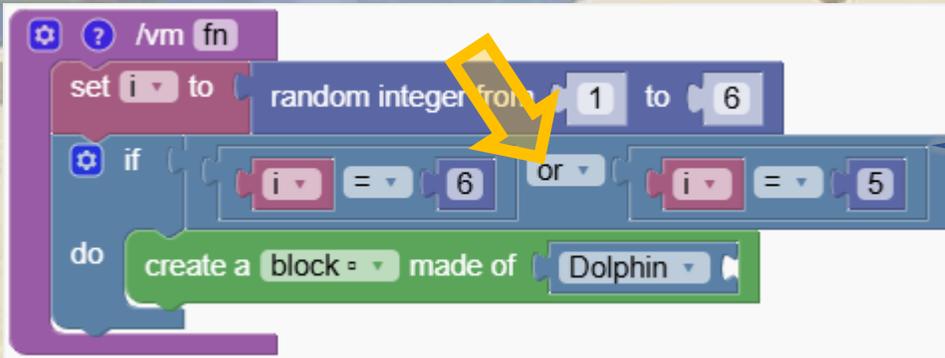


```
set i to random integer from 1 to 6
if i != 6
do create a block made of Dolphin
```

Create a dolphin only if the variable "i" does NOT contain the value 6

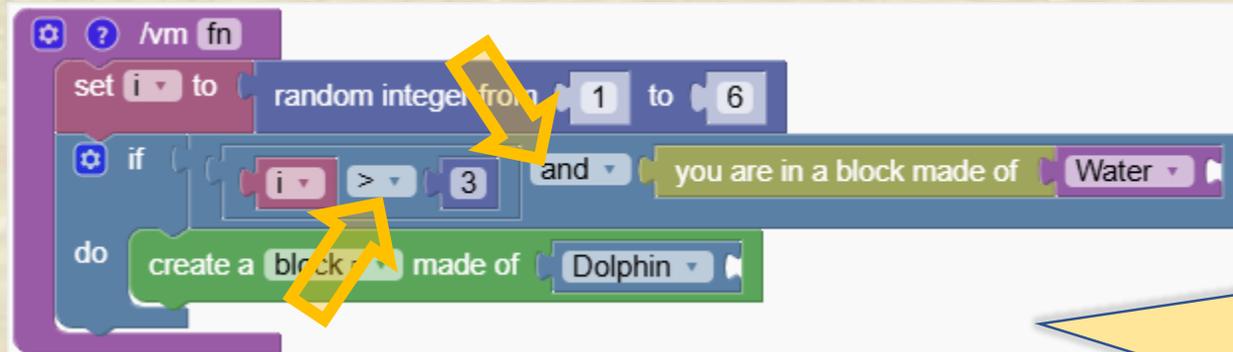
The Logic Blocks

The conditions can be combined



```
function fn {
  set i to random integer from 1 to 6
  if i = 6 or i = 5
  do create a block made of Dolphin
}
```

Create a dolphin only if the variable 'i' contains the value 5 OR 6

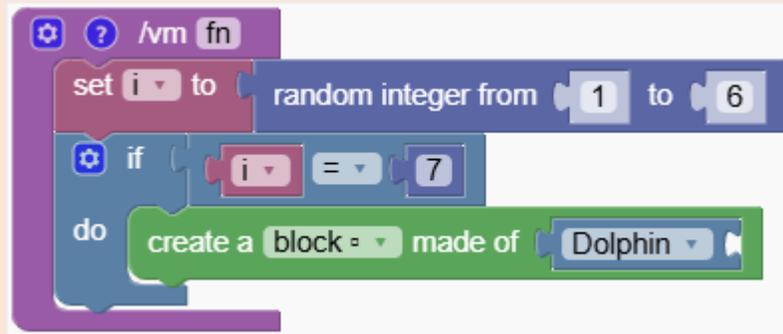


```
function fn {
  set i to random integer from 1 to 6
  if i > 3 and you are in a block made of Water
  do create a block made of Dolphin
}
```

Create a dolphin only if the variable 'i' contains a value bigger than 3 and the robot is in a block made of water

Error in the Conditions

Identify and fix errors in conditions that may make no sense or result in incorrect logic.



```
function (/vm fn)
  set i to random integer from 1 to 6
  if i = 7
    do
      create a block made of Dolphin
```

The code block shows a function that sets a variable `i` to a random integer between 1 and 6. It then checks if `i` is equal to 7. Since `i` can only be between 1 and 6, this condition will never be true, and the block will never execute the `do` part.



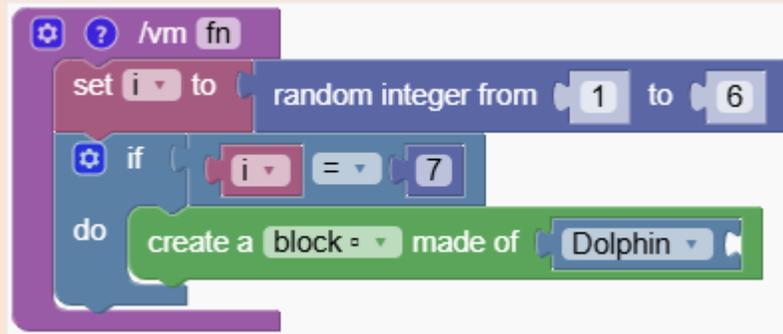
```
function (/vm fn)
  set i to random integer from 1 to 6
  if i = 3 and i = 5
    do
      create a block made of Dolphin
```

The code block shows a function that sets a variable `i` to a random integer between 1 and 6. It then checks if `i` is equal to 3 and `i` is equal to 5. Since `i` can only be one value at a time, it cannot be both 3 and 5, so this condition will never be true, and the block will never execute the `do` part.

Quiz

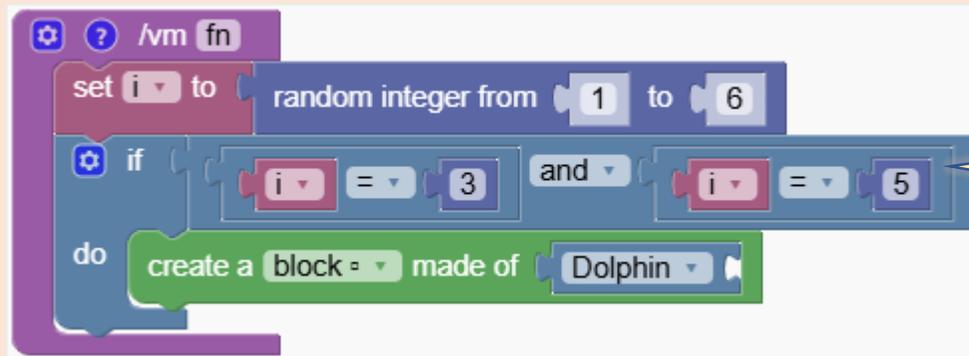
Error in the Conditions

Identify and fix errors in conditions that may make no sense or result in incorrect logic.



```
function fn() {
  set i to random integer from 1 to 6
  if i = 7
  do create a block made of Dolphin
}
```

The variable 'i' cannot contain the value 7



```
function fn() {
  set i to random integer from 1 to 6
  if i = 3 and i = 5
  do create a block made of Dolphin
}
```

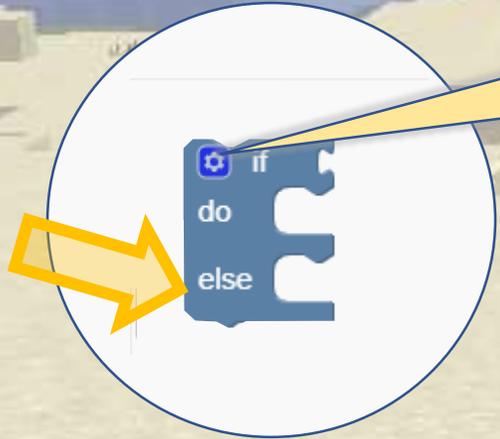
The variable 'i' cannot contain the value 3 and 5 at the same time

Quiz

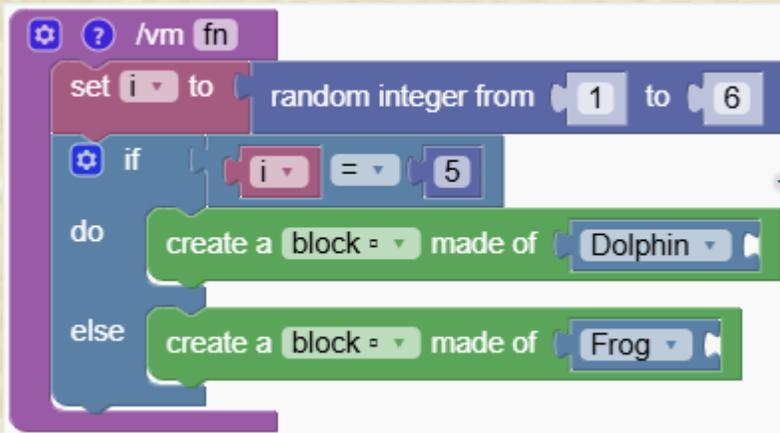
The Logic Blocks

The 'if else' block allows for defining an alternative action

Click here to open the menu to customize the block

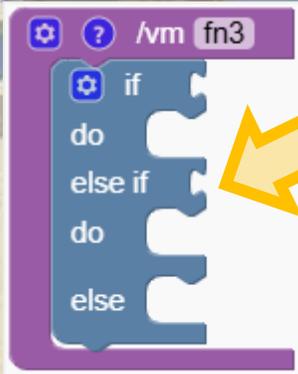


If 'i' contains the number 5 then spawn a dolphin, for all other values spawn a frog



The Logic Blocks

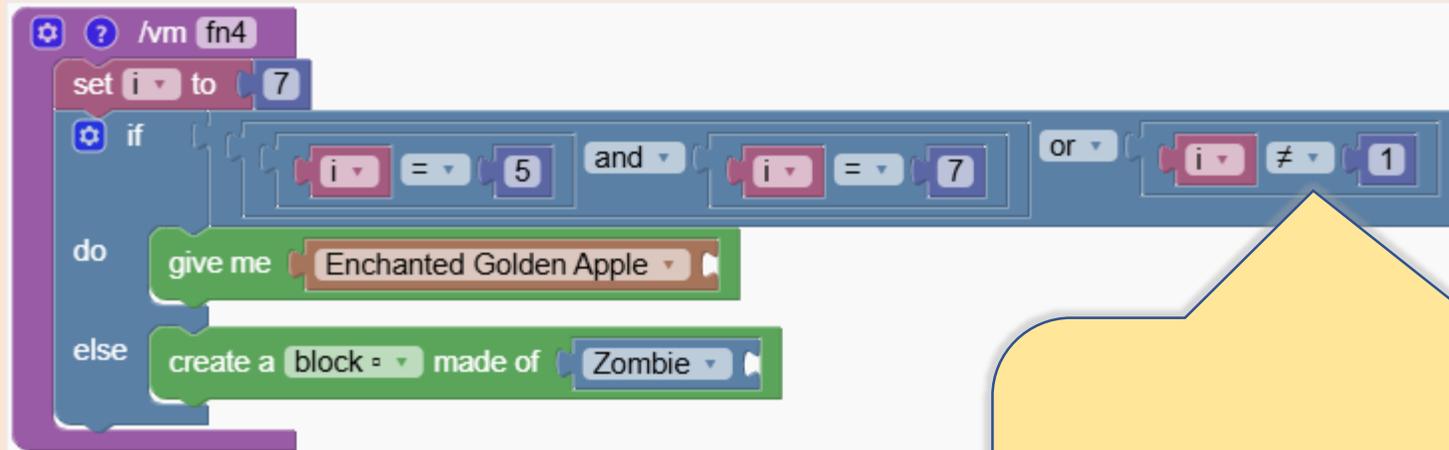
You can customize the block to give more alternatives



Try your luck!
Running this program you could get a prize or be attacked by a zombie

What's the Result of the Program?

If I run this program what will happen?

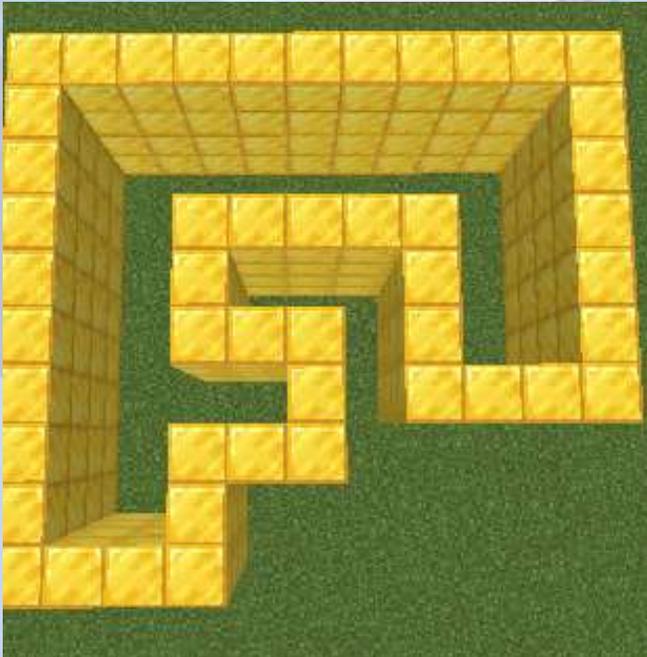


```
set i to 7
if (i = 5 and i = 7 or i ≠ 1)
do
give me Enchanted Golden Apple
else
create a block made of Zombie
```

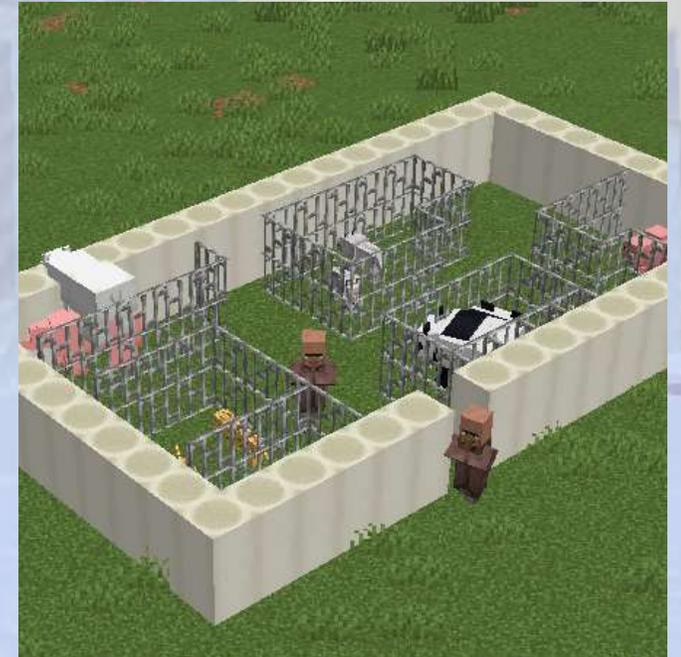
**We get the apple !
The variable 'i' doesn't contain the
value 1**

Quiz

Complex Shapes



Create non-geometric shapes



Complex shapes

 Section Overview

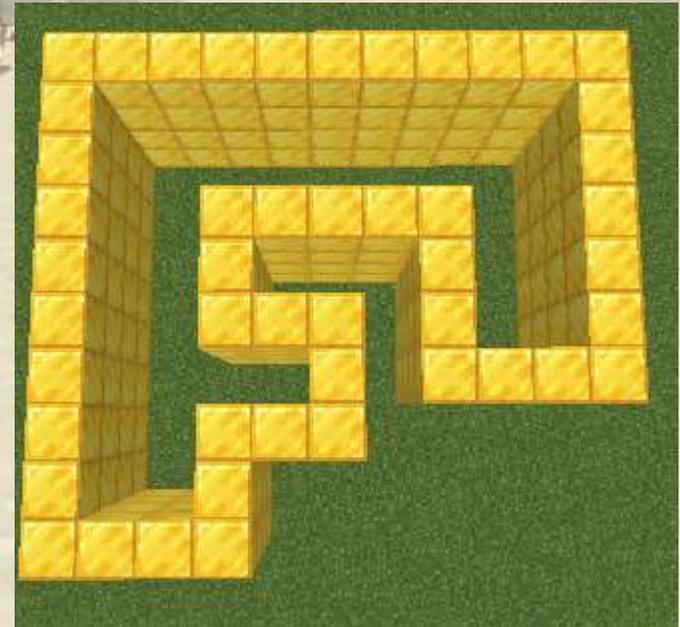
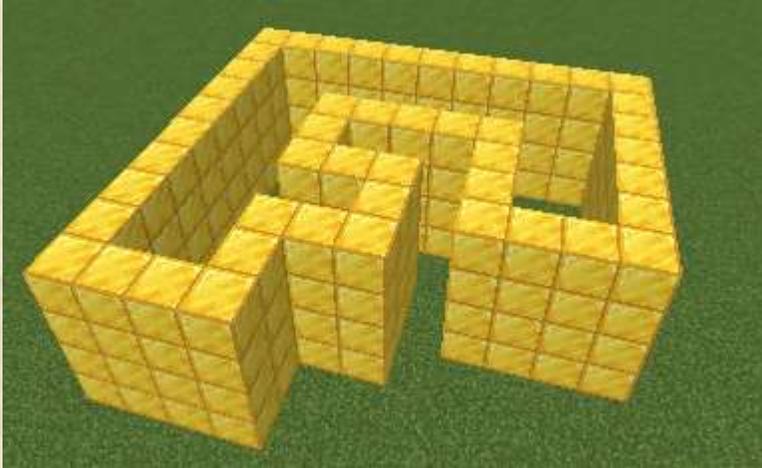
 Objectives

In this section, we will learn how to create non-geometric shapes, from drawing simple to more complex structures.

 Expected Outcomes

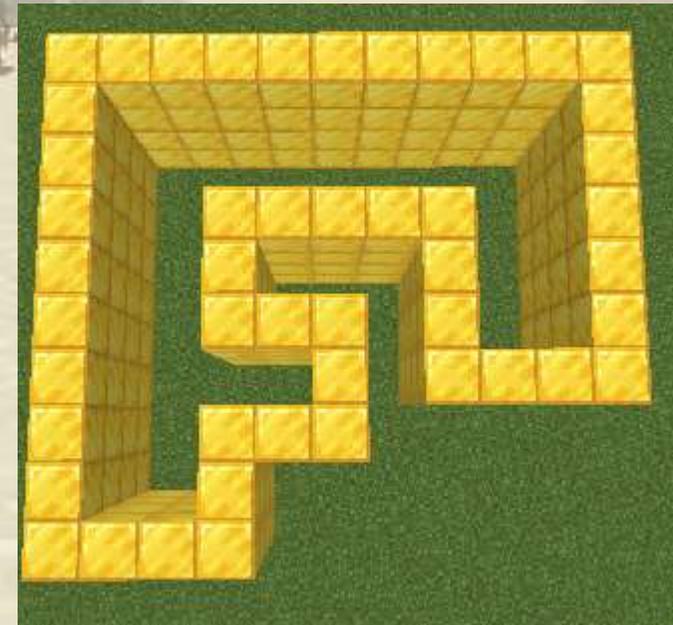
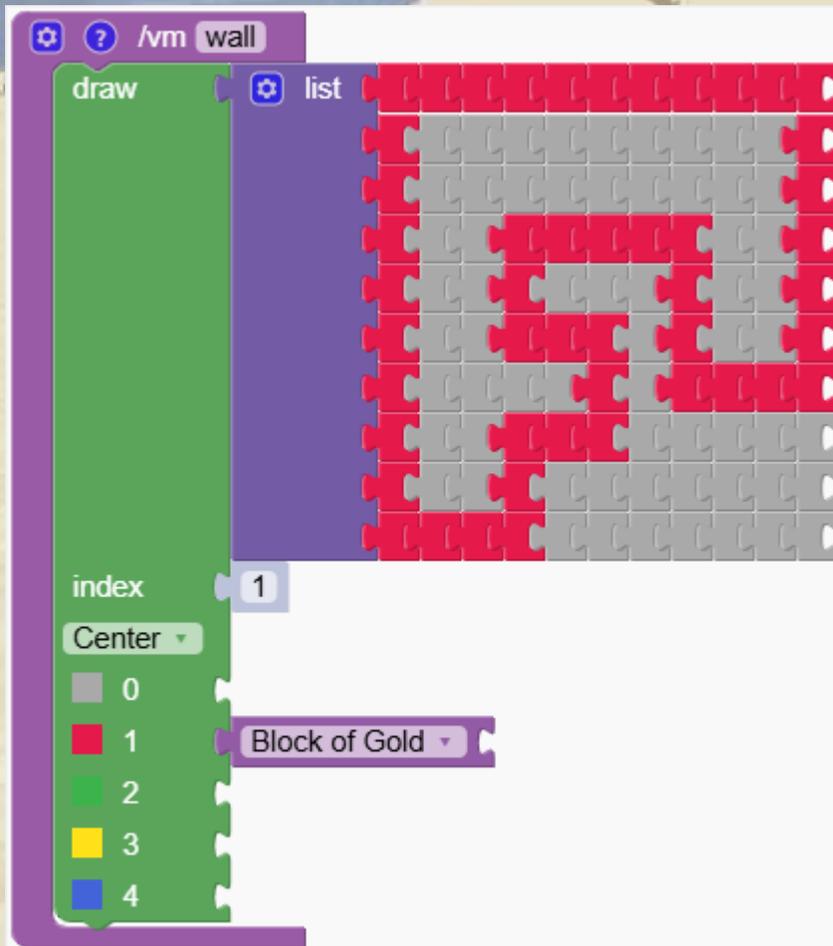
What are we going to learn

We learn how to make shapes that aren't geometric, allowing us to create more creative and freeform structures.



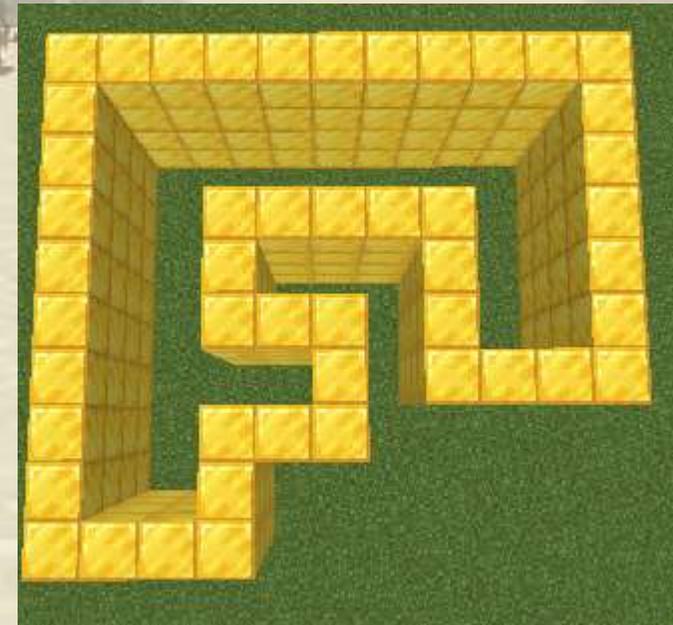
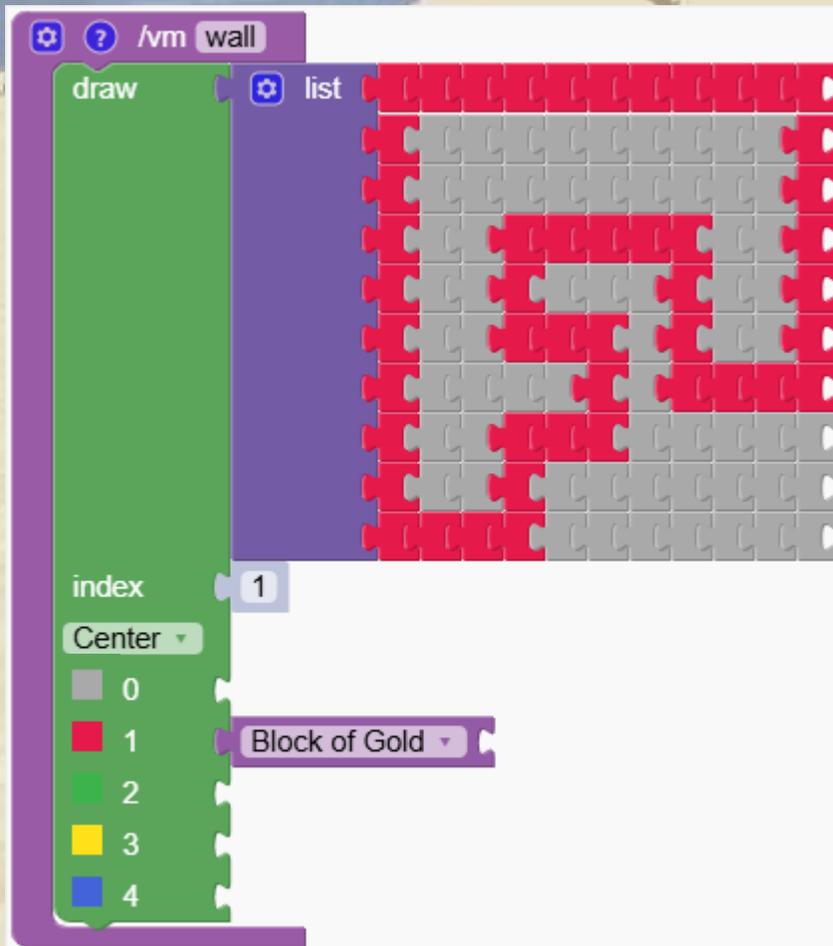
The drawing block

We show what the drawing block is and how it works, forming the basis for creating custom shapes. Look at the program below. We can draw exact images for our programs



The drawing block

We show what the drawing block is and how it works, forming the basis for creating custom shapes. Look at the program below. We can draw exact images for our programs



The drawing block

Press a key to color the grey blocks.

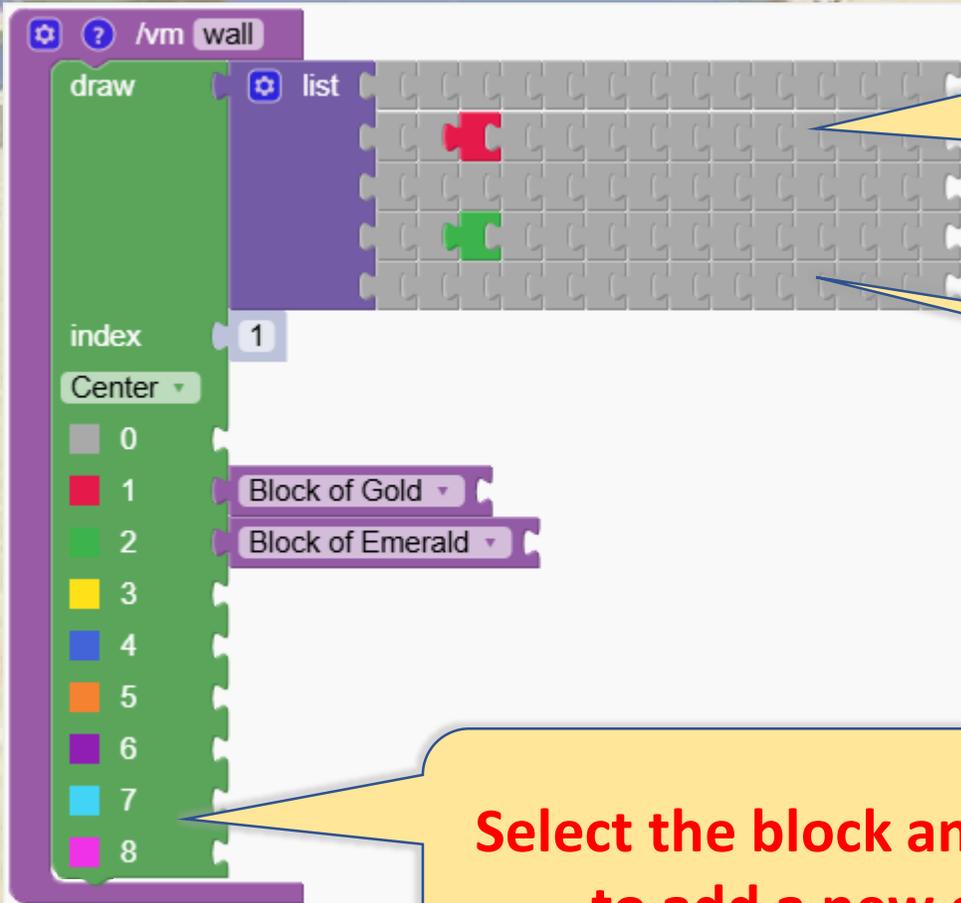
The image shows a Scratch code editor window. At the top, there is a purple bar with a gear icon, a question mark icon, and the text '/vm wall'. Below this is a green 'draw' block. Attached to the right side of the 'draw' block is a purple 'list' block. Inside the 'list' block, there is a grey '1' block. Below the 'draw' block, there is a color palette with a dropdown menu set to 'Center'. The palette has five options: 0 (grey), 1 (red), 2 (green), 3 (yellow), and 4 (blue). Below the color palette, there is a purple 'Block of Gold' block attached to the bottom of the 'draw' block.

Select a block and the press '1' to color it red.
Below, we attached a block of gold to indicate
that red means gold

The drawing block

We can add or remove rows and columns by using keyboard keys 'i' and 'l'.

We can also add more colors using keyboard key 'i'.



Select a block and press 'i' to add a column or 'l' to add a row

Right click to see all options

Select the block and press 'i' to add a new colors

The drawing block

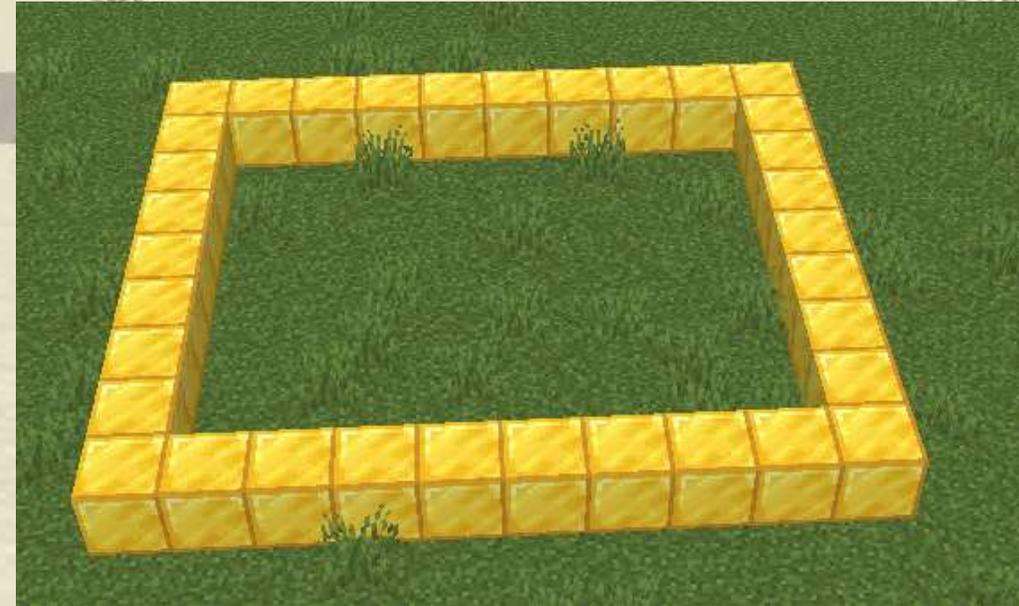
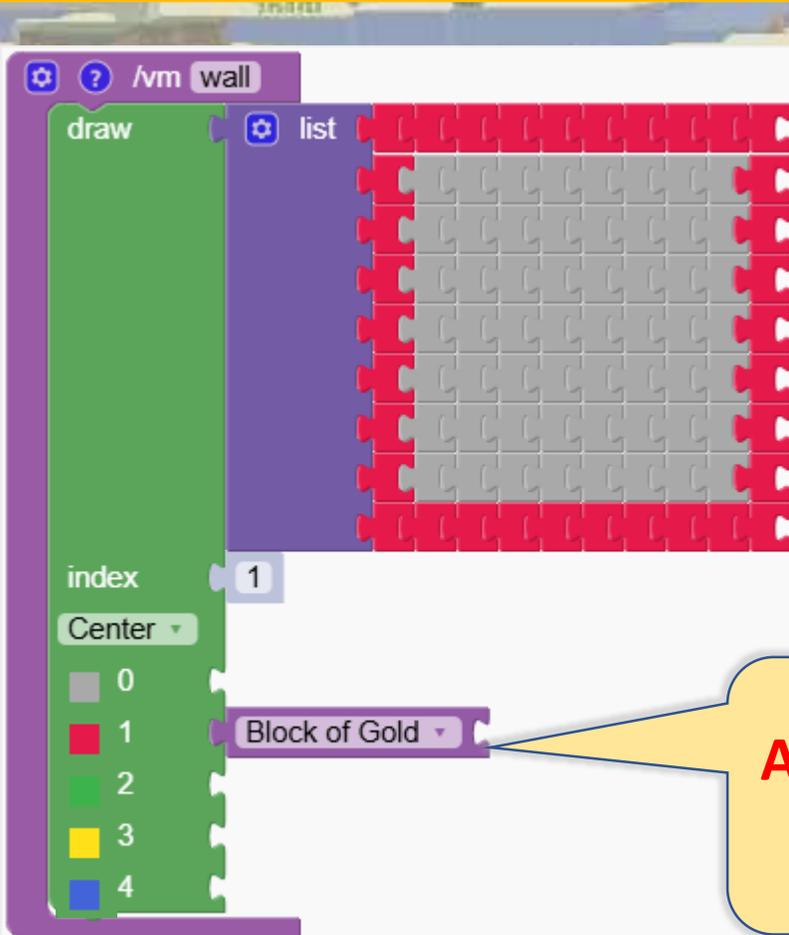
To automatically draw a rectangle follow the shown steps below
(If you right-click you have all the options)

The image shows two screenshots of a game engine's drawing interface. The left screenshot shows a 'draw' block with a 'list' block attached. A red square is highlighted at the top-left corner of a grey rectangular area. A yellow callout bubble points to this red square with the text: "First select the top left corner of the rectangle and change its color". The right screenshot shows the same interface, but the bottom-right corner of the grey area is now highlighted in red. A yellow callout bubble points to this red square with the text: "Then select the bottom right block and press 'b'". Below the screenshots is a legend for the 'index' property, showing a dropdown menu set to 'Center' and a list of indices: 0 (grey), 1 (red), 2 (green), 3 (yellow), and 4 (blue).

index
Center ▾
0
1
2
3
4

The drawing block

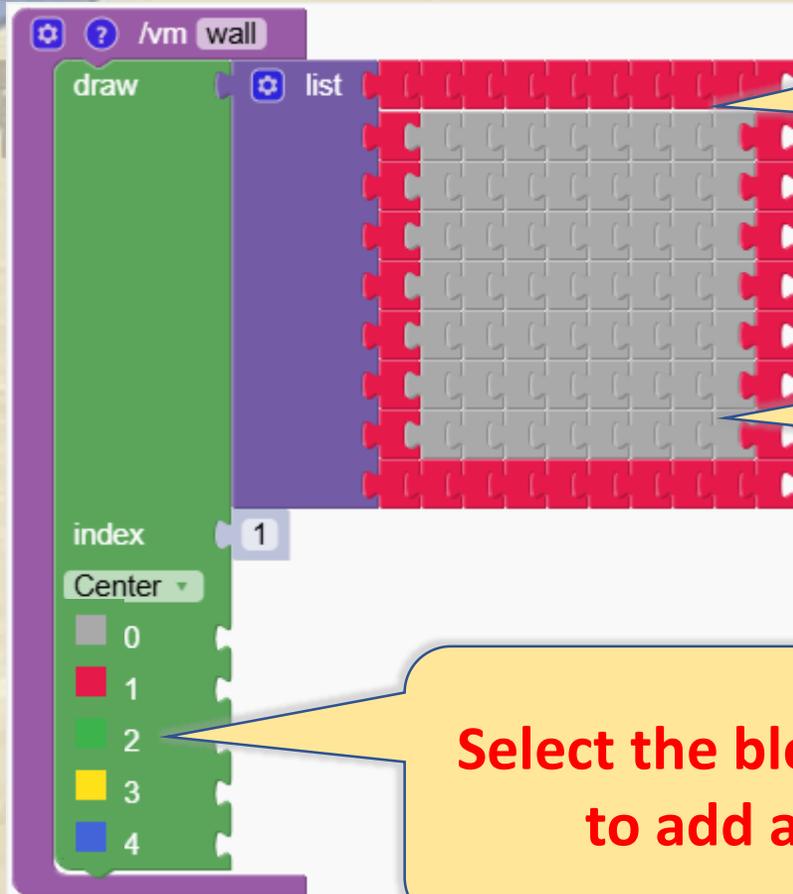
We show what the drawing block is and how it works, forming the basis for creating custom shapes. Look at the program below. We can draw exact images for our programs



Add a block of gold. '1' is now a block of gold

The drawing block

We show what the drawing block is and how it works, forming the basis for creating custom shapes. Look at the program below. We can draw exact images for our programs



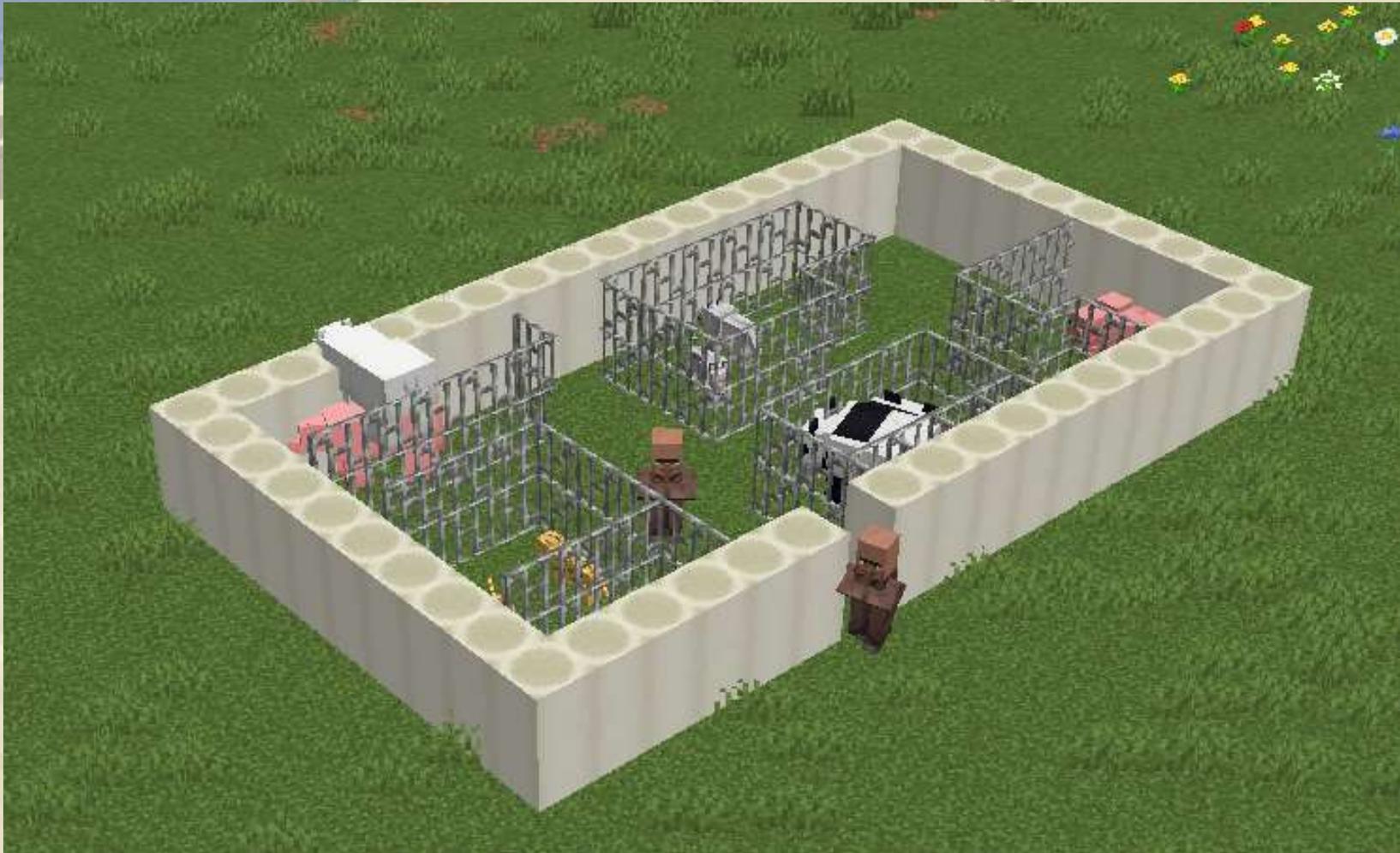
Select a block and press 'i' to add a column or 'l' to add a row

Right click to see all options

Select the block and press 'i' to add a new color

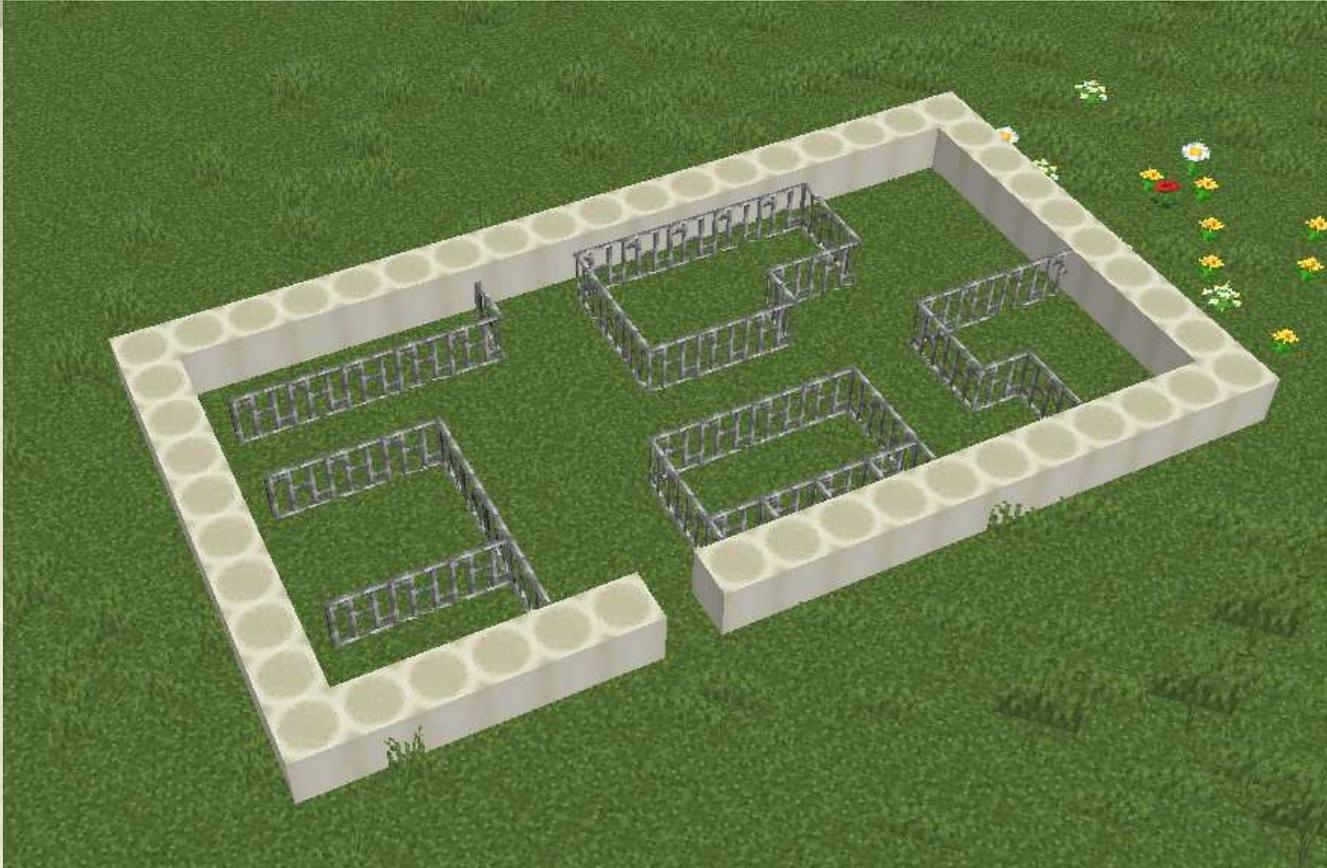
⚡ Zoo

Let's make a zoo!



⚡ Zoo

First we create the outside wall and the cages



⚡ Zoo

Then we add the animals.



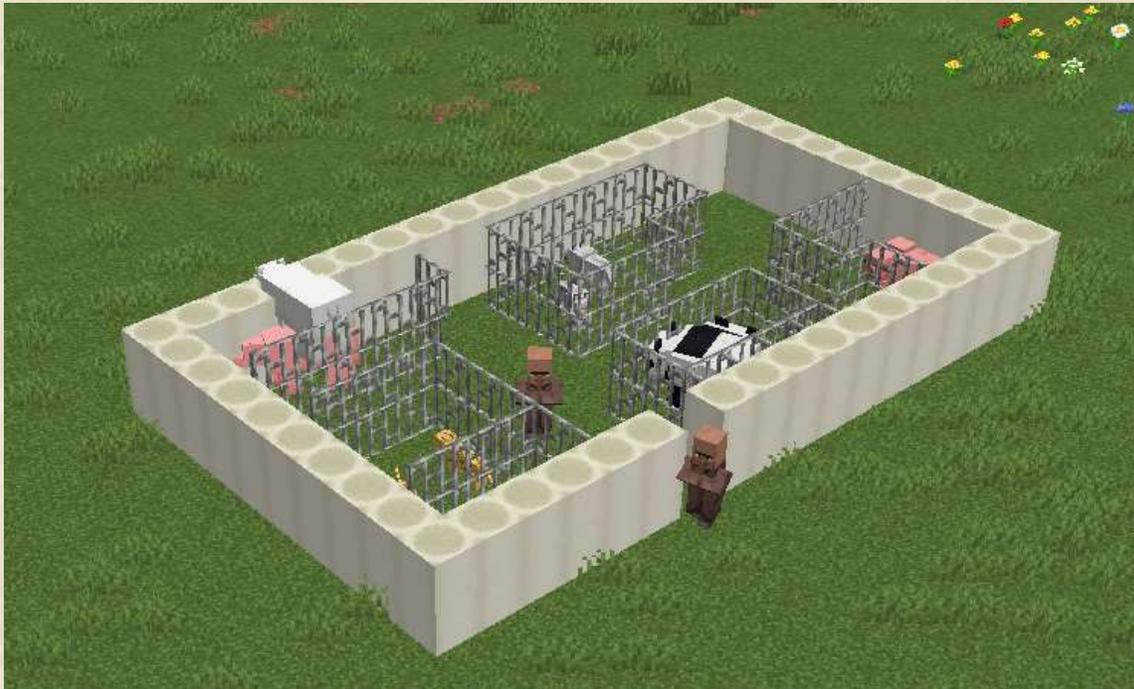
Code editor for a Minecraft world named 'zoo'. The code is written in a block-based language and includes a 'draw' block and a 'list' block.

The 'draw' block is connected to the 'list' block. The 'list' block contains a list of items, each with a color-coded index and a corresponding animal name:

| index | Animal |
|-------|------------|
| 0 | Air |
| 1 | Iron Bars |
| 2 | Bone Block |
| 3 | Chicken |
| 4 | Pig |
| 5 | Ocelot |
| 6 | Wolf |
| 7 | Polar Bear |
| 8 | Panda |
| 9 | Villager |

⚡ Zoo

To stop the animals from escaping, we make the cages one block higher by repeating the drawing.

A screenshot of a Minecraft command block interface. The command block is set to 'repeat' mode with a count of '2'. The 'do' block contains a 'draw' block with a 'list' block. The 'list' block contains a list of items with their corresponding colors: 0 (Air), 1 (Iron Bars), 2 (Bone Block), 3 (Chicken), 4 (Pig), 5 (Ocelot), 6 (Wolf), 7 (Polar Bear), 8 (Panda), and 9 (Villager). The 'draw' block is set to 'index 1' and 'Left'. The 'go' block is set to '1 blocks up 1'.

repeat 2 times

do draw list

index 1

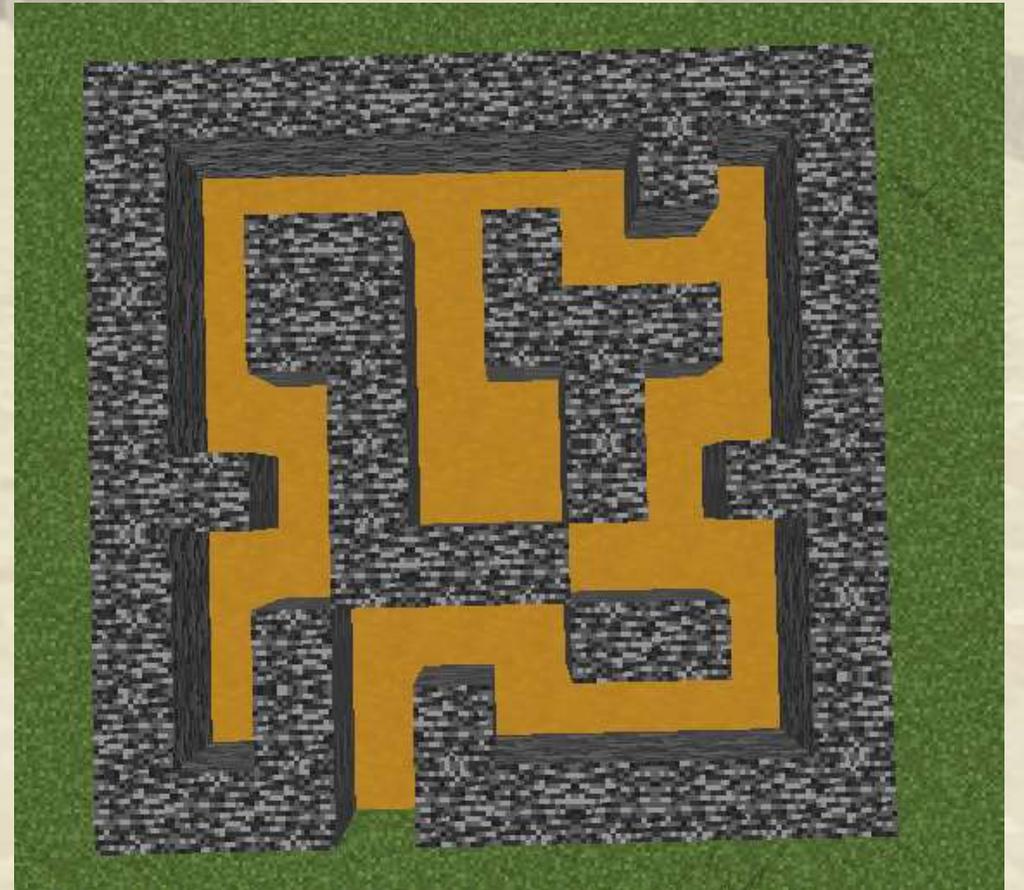
Left

- 0 Air
- 1 Iron Bars
- 2 Bone Block
- 3 Chicken
- 4 Pig
- 5 Ocelot
- 6 Wolf
- 7 Polar Bear
- 8 Panda
- 9 Villager

go 1 blocks up 1

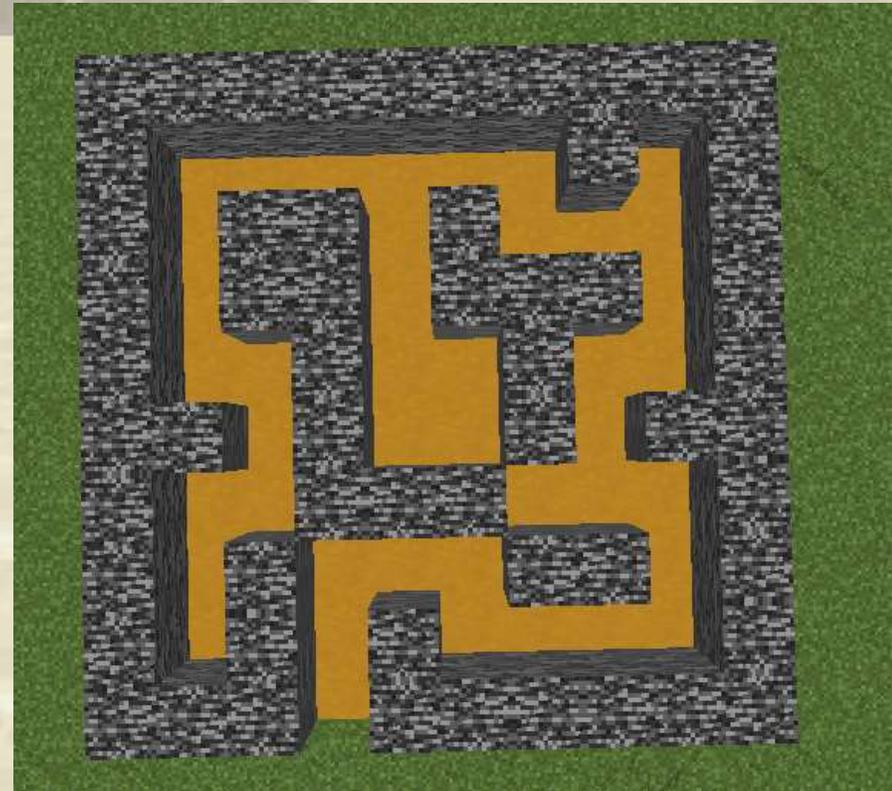
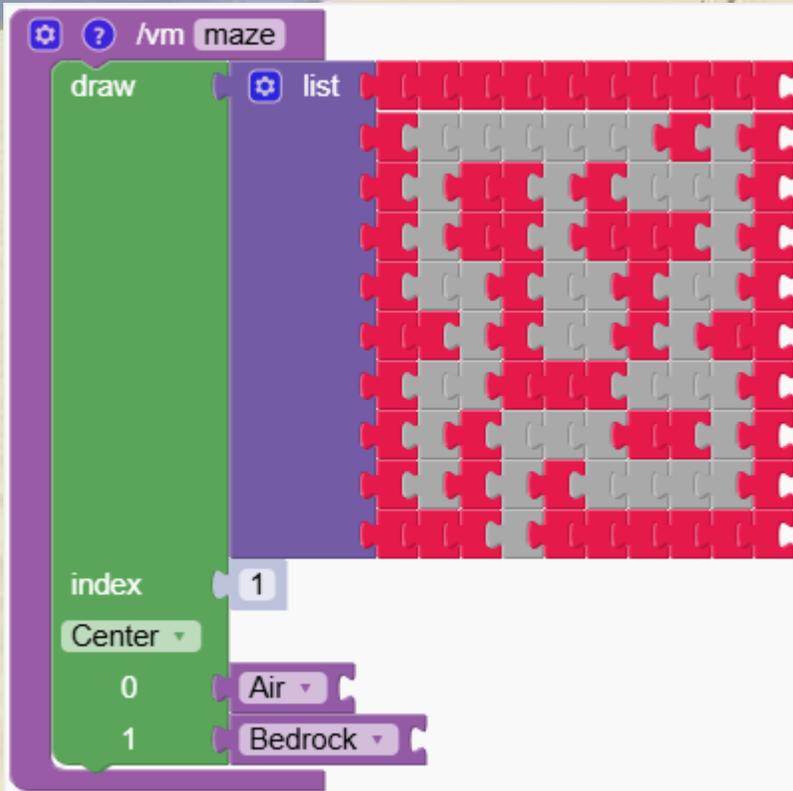
Fun ⚡ Design a maze

Design your maze and invite others to try and walk through it.



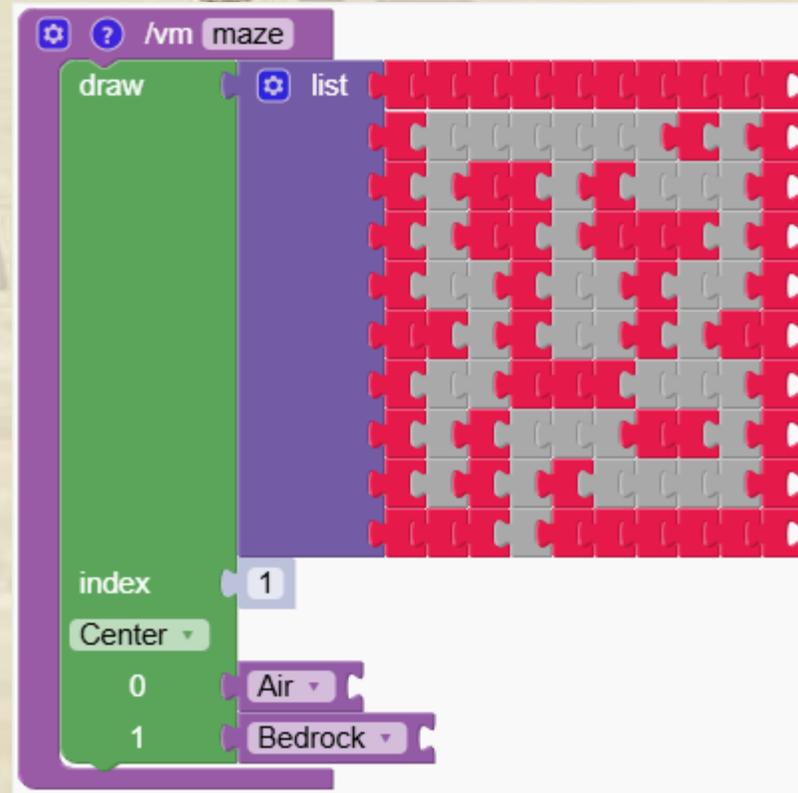
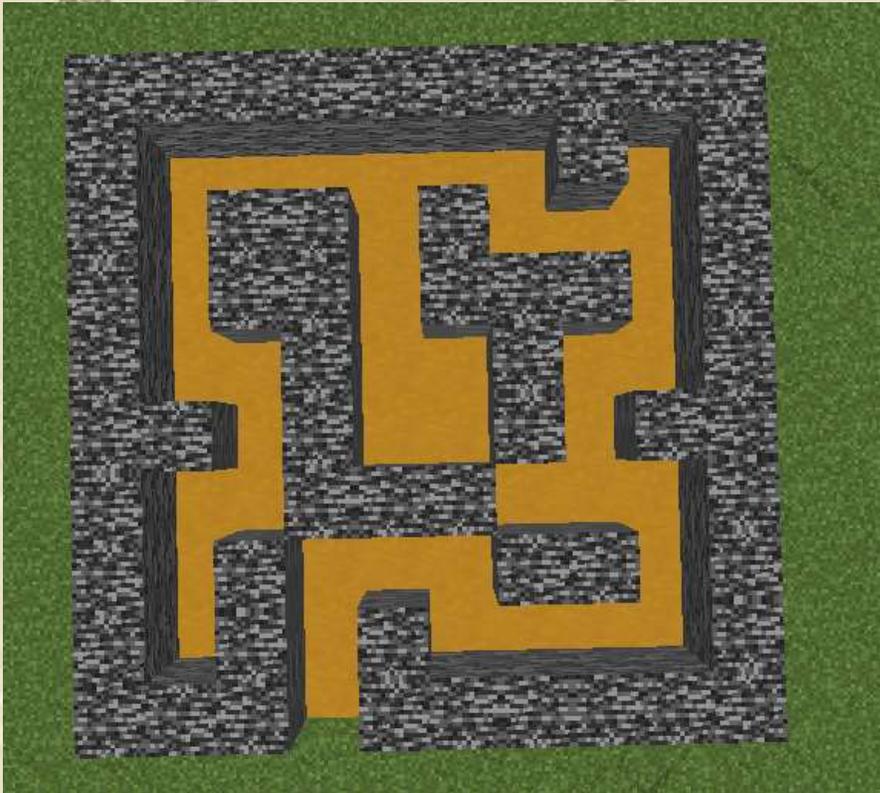
Fun ⚡ Design a maze

First we make the basic layout.



Fun ⚡ Design a maze

First we make the basic layout.



Fun ⚡ Design a maze

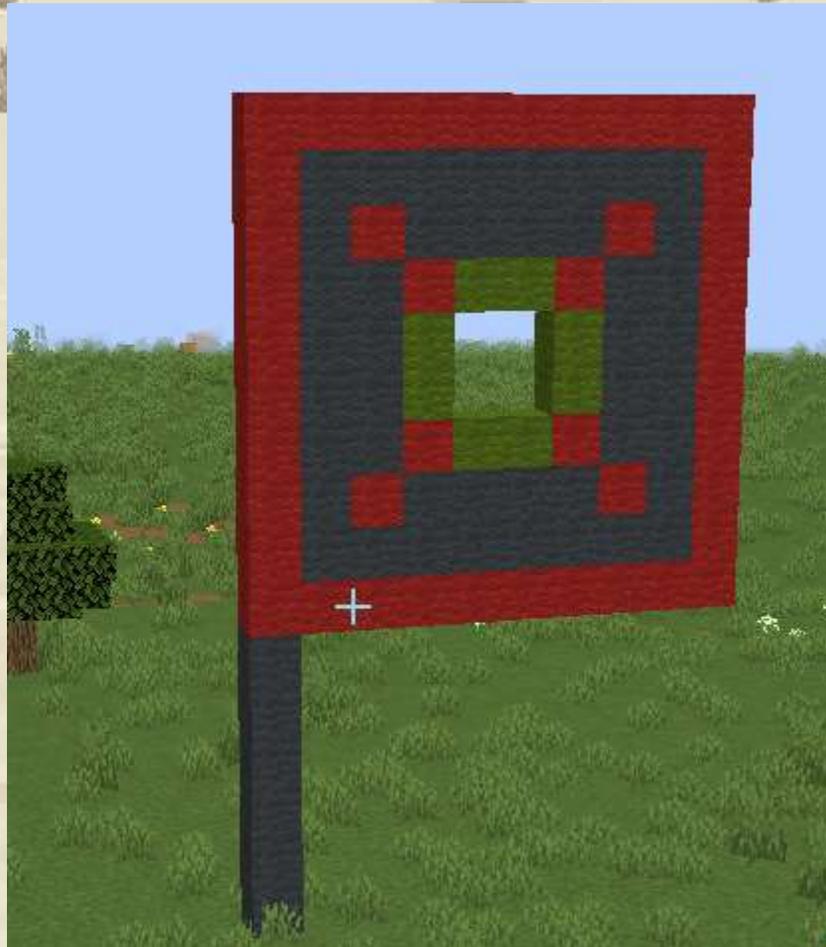
Then we add a floor and a roof. The walls are repeated 3 times



```
create a full square of width 10 made of Yellow Terracotta
go 1 blocks up 1
repeat 3 times
do draw
  list
  index 1
  Center
  0 Air
  1 Bedrock
go 1 blocks up 1
create a full square of width 10 made of Glass
```

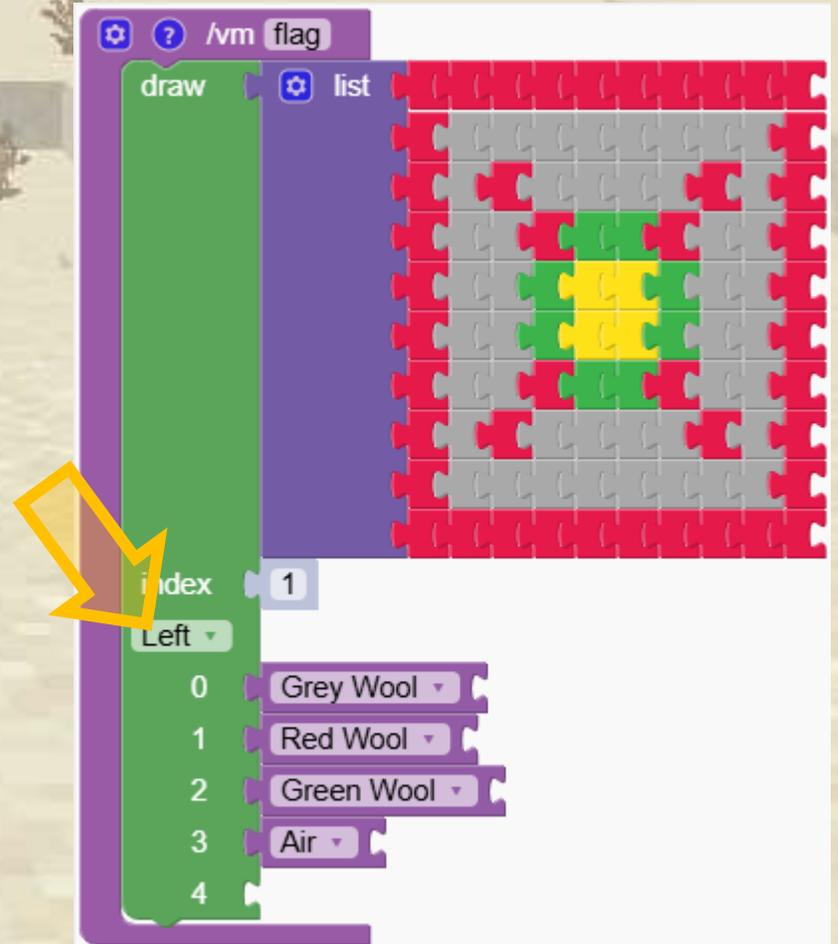
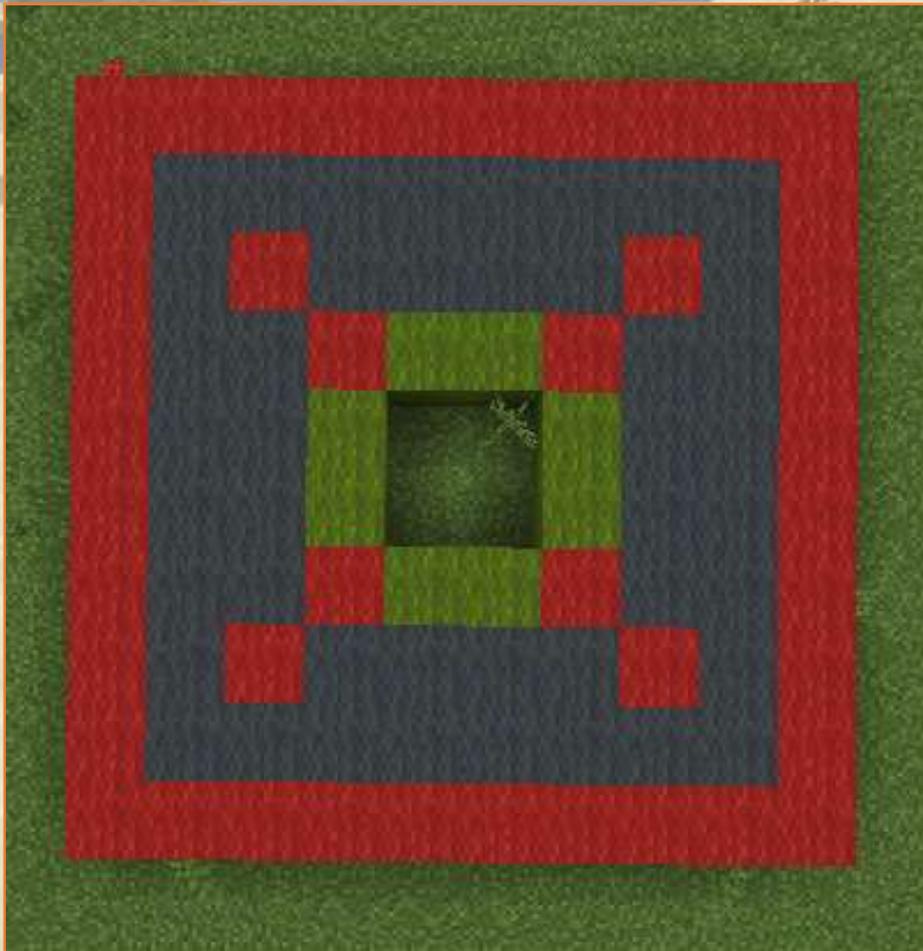
⚡ Make your own flag

Learn how to create a custom flag, using different shapes to form the design.



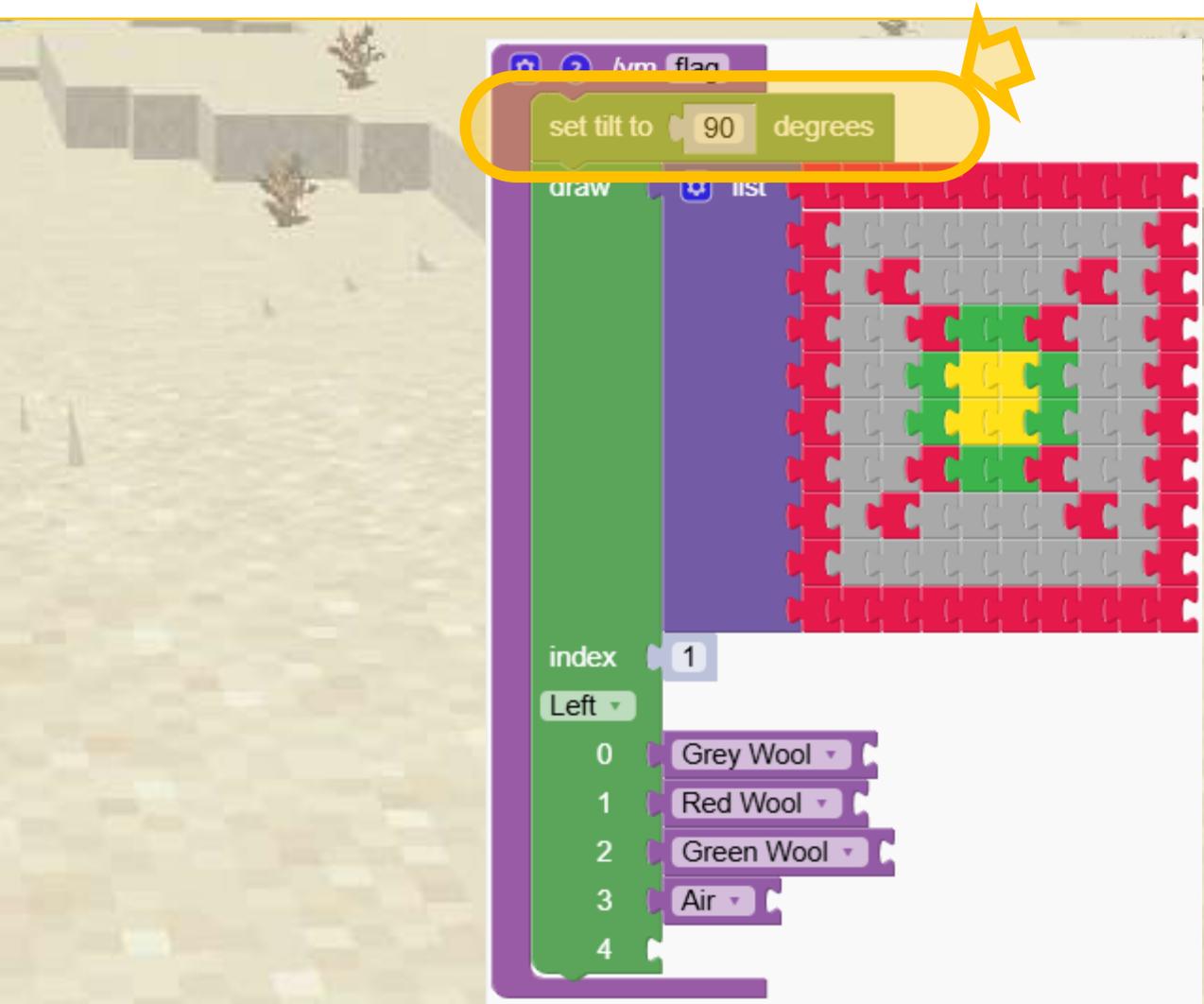
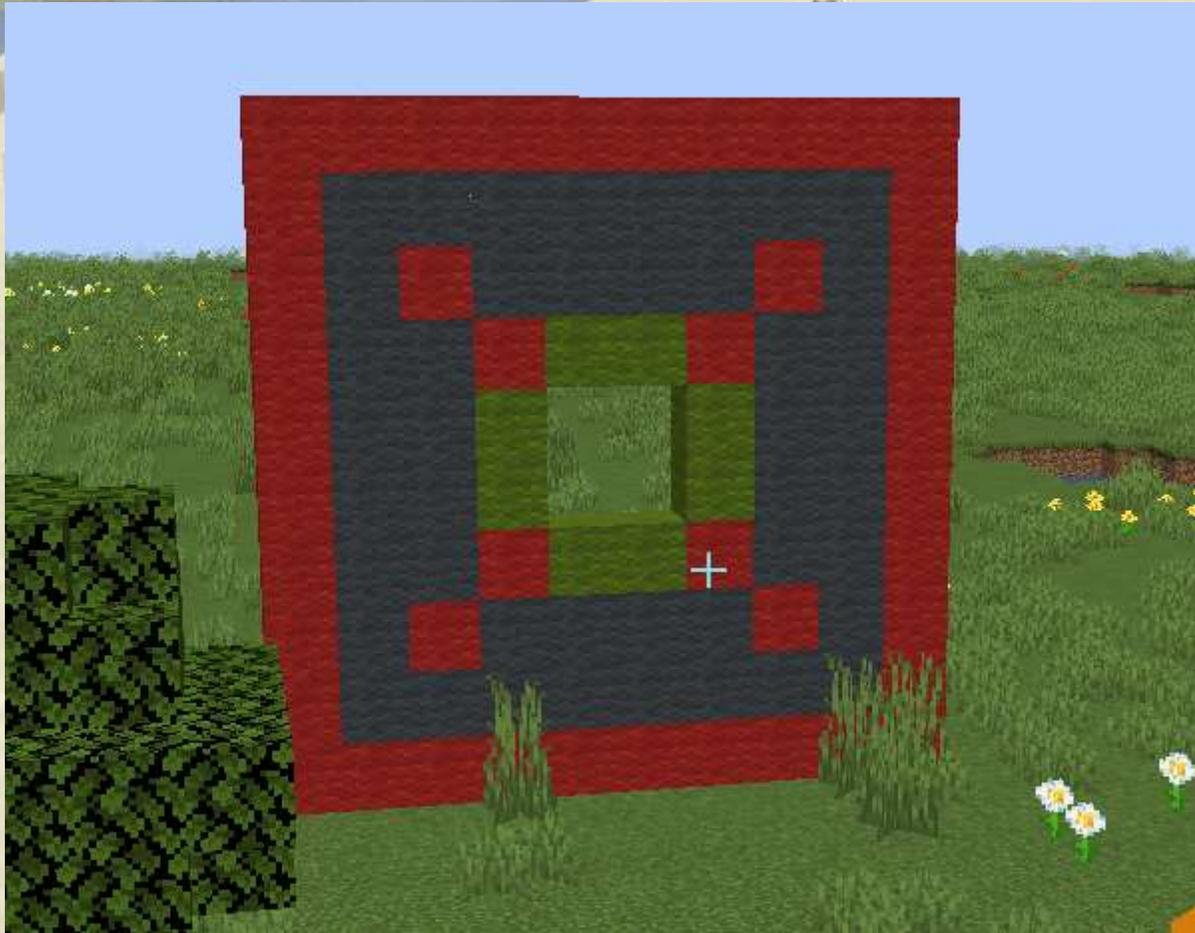
⚡ Make your own flag

First we create our design on the ground. Notice that we set the starting point to left instead of center



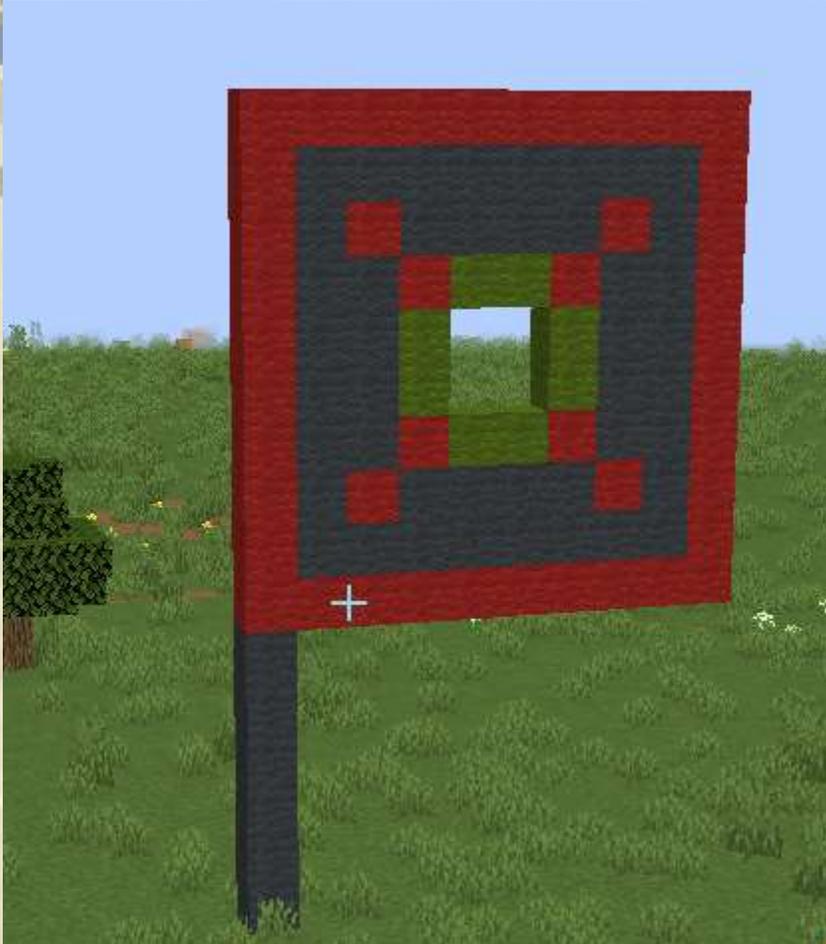
⚡ Make your own flag

We change the tilt to make the image vertical

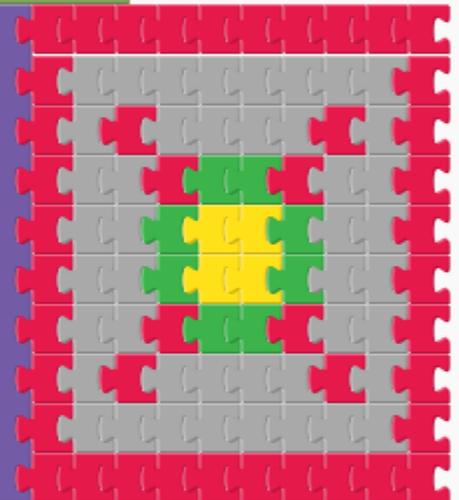


⚡ Make your own flag

We add a simple base made of 6 blocks



```
repeat 6 times
do
  create a block = made of Grey Wool
  go 1 blocks up 1
set tilt to 90 degrees
draw list
index 1
Left
0 Grey Wool
1 Red Wool
2 Green Wool
3 Air
4
```



Make a House

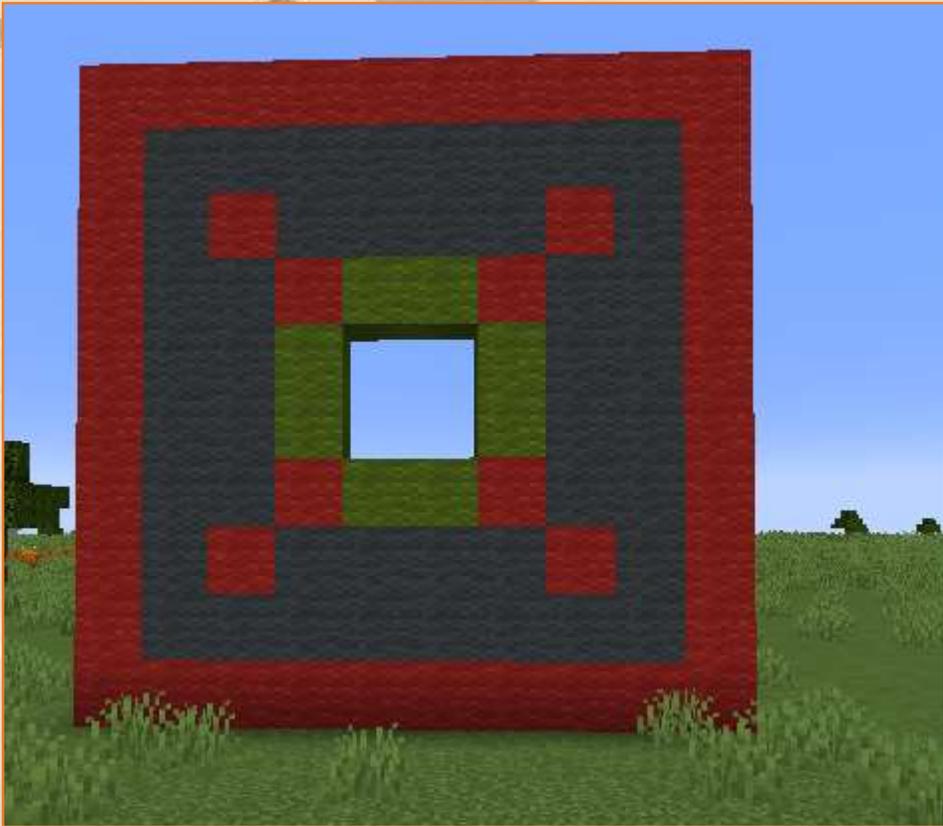
Learn how to position drawings to create a cubic house, practicing the basics of 3D shapes.



⚡ Make a House

First we create a simple wall.

Notice that we reset the tilt of the wall.



set tilt to 90 degrees

draw

list

index 1

Left

- 0 Grey Wool
- 1 Red Wool
- 2 Green Wool
- 3 Air

set tilt to 0 degrees

⚡ Make a House

The following program paints 4 times the wall.

(The program calls the function “wall” that we created before)

Let see how it works:



```

/vm cube
go 15 blocks forward ↑
repeat 4 times
do
  wall
  go 9 blocks right →
  turn right by 90 degrees

```

Tell robot to go further away

⚡ Make a House

We create the first wall

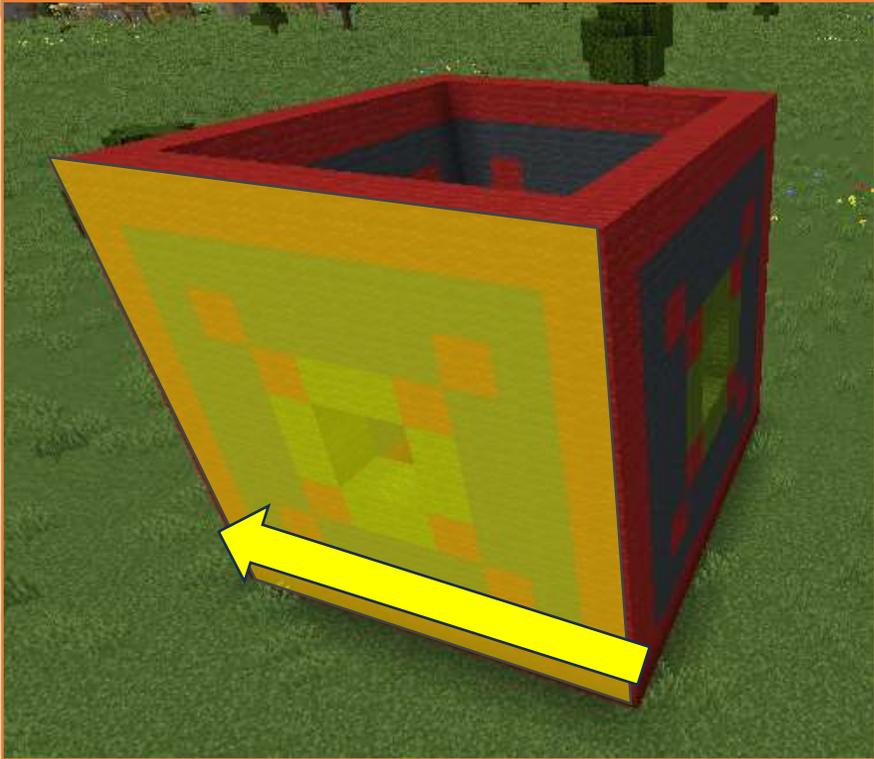


```
vm cube
go 15 blocks forward ↑
repeat 4 times
do wall
  go 9 blocks right →
  turn by 90 degrees
```

Makes the front wall

⚡ Make a House

We tell robot to move to the right so the next wall will start at the right point



```

? /vm cube
go 15 blocks forward ↑
repeat 4 times
do
  wall
  go 9 blocks right →
  turn right by 90 degrees

```

Move robot to the right

⚡ Make a House

We turn the robot in the right direction.



```

? /vm cube
go 15 blocks forward ↑
repeat 4 times
do
  wall
  go 9 blocks right →
  turn right by 90 degrees

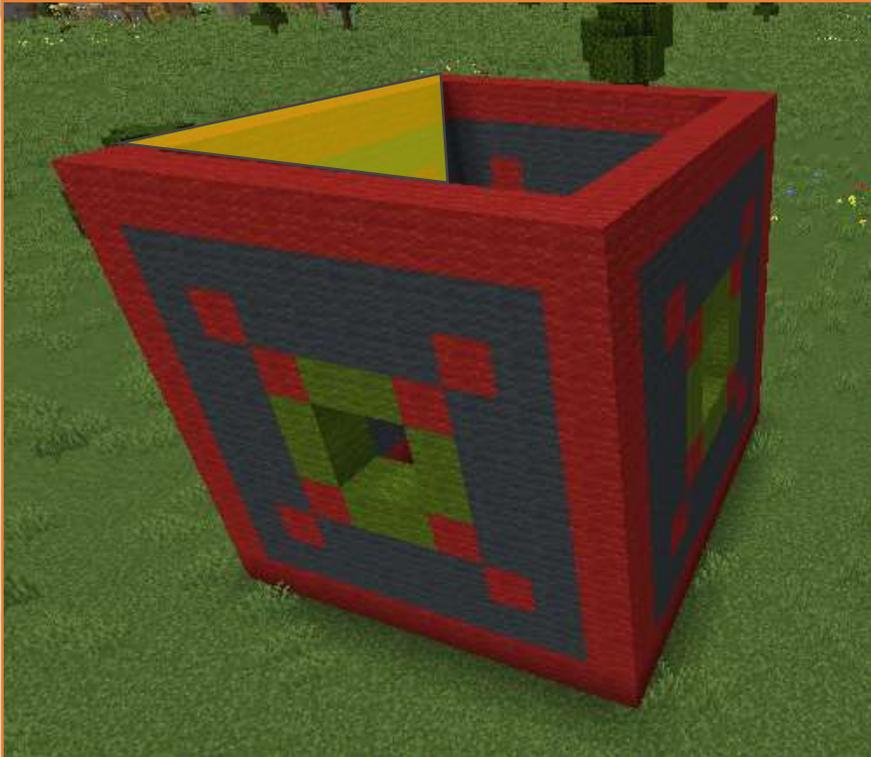
```

Turn robot to the right

⚡ Make a House

We paint the next wall.

We repeat this process 4 times



```
vm cube
go 15 blocks forward ↑
repeat 4 times
do wall
  go 9 blocks right →
  turn by 90 degrees
```

Paint the side wall

⚡ The Colosseum

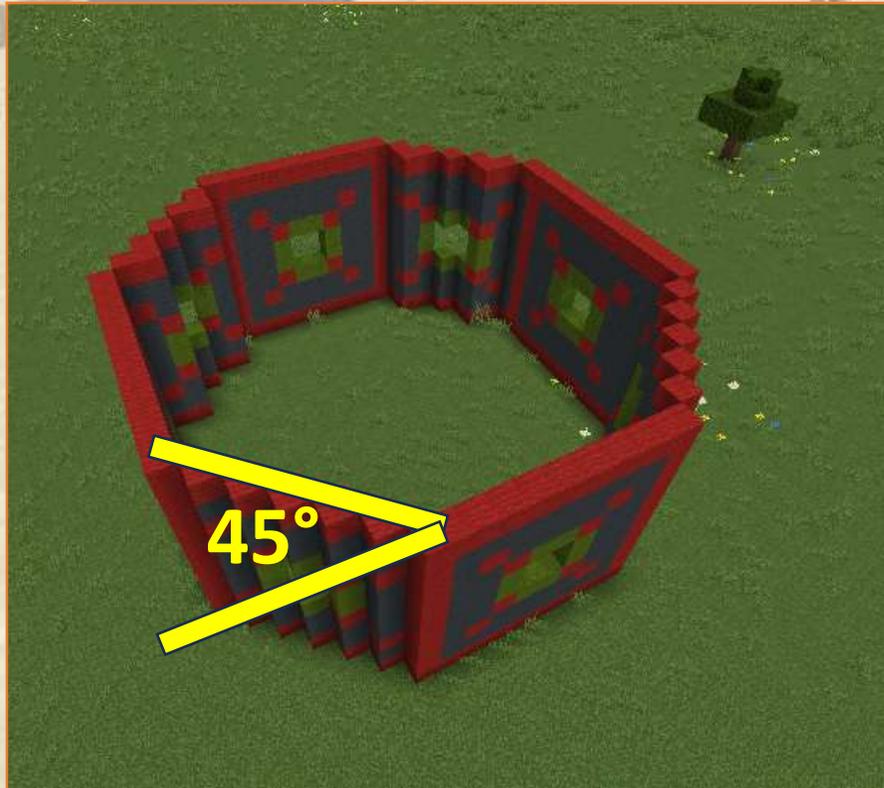
With the help of some math, we can transform the house into a colosseum structure



⚡ The Colosseum

We extend the house program to make it into a colosseum.

We tell robot to turn only 45 degrees so that the wall can be repeated 8 times. ($8 \times 45 = 360$)



```
lvn cube
go 15 blocks forward ↑
repeat 8 times
do
  wall
  go 9 blocks right →
  turn right by 45 degrees
```

Repeat 8 times

Turn 45 degrees

⚡ The Colosseum

Now we make 3 times more walls. Just adjust the rotation to 15 degrees ($24 \times 15 = 360$)



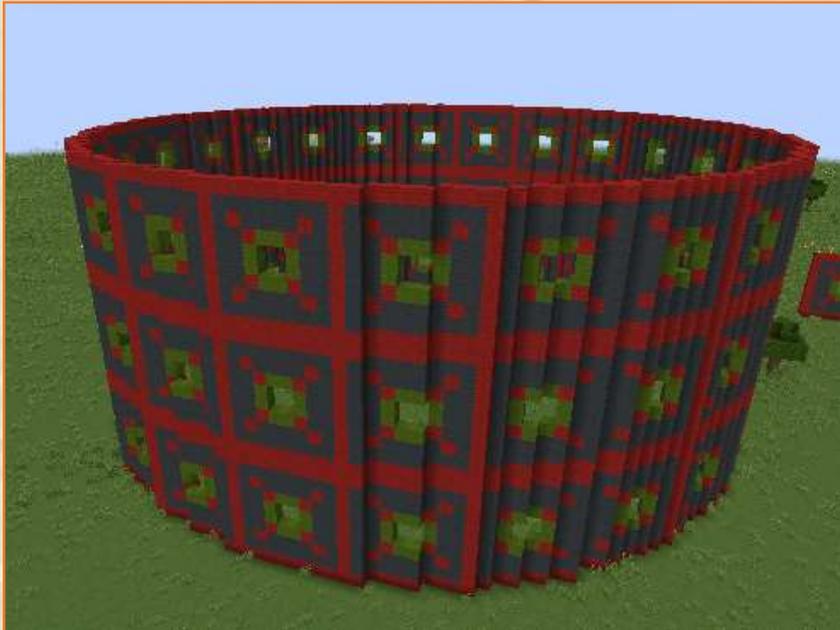
```
vm cube
go 15 blocks forward ↑
repeat 24 times
do
  wall
  go 9 blocks right →
  turn right by 15 degrees
```

Repeat 24 times

Turn 15 degrees

⚡ The Colosseum

Let's repeat it 3 times



```
vm cube
go 15 blocks forward ↑
repeat 3 times
do
  repeat 24 times
  do
    wall
    go 9 blocks right →
    turn right by 15 degrees
  go 10 blocks up ↑
```

3 layers

The next layer is 10 blocks above the previous one

⚡ The Mushroom House

Rotate drawings to create unique and fantastical house designs, like a mushroom house.



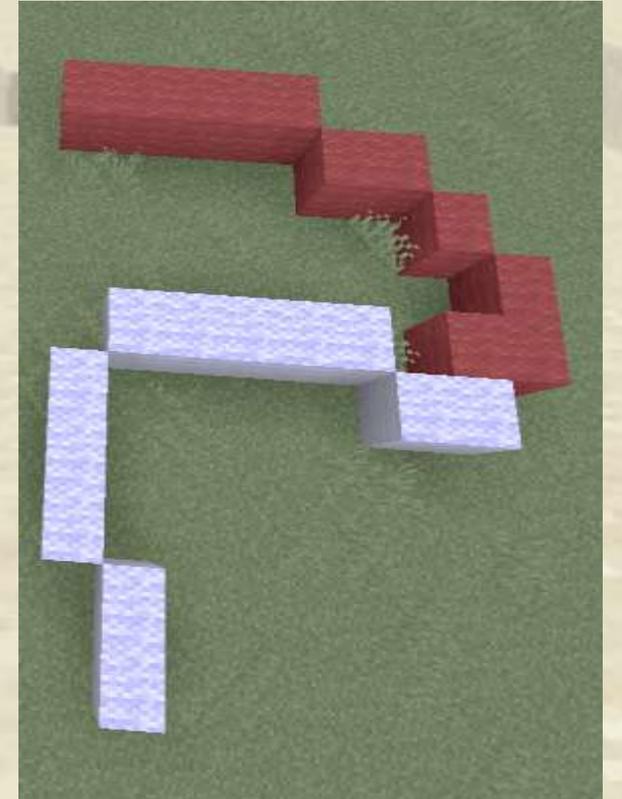
⚡ The Mushroom House

We first create half a slice of our mushroom house



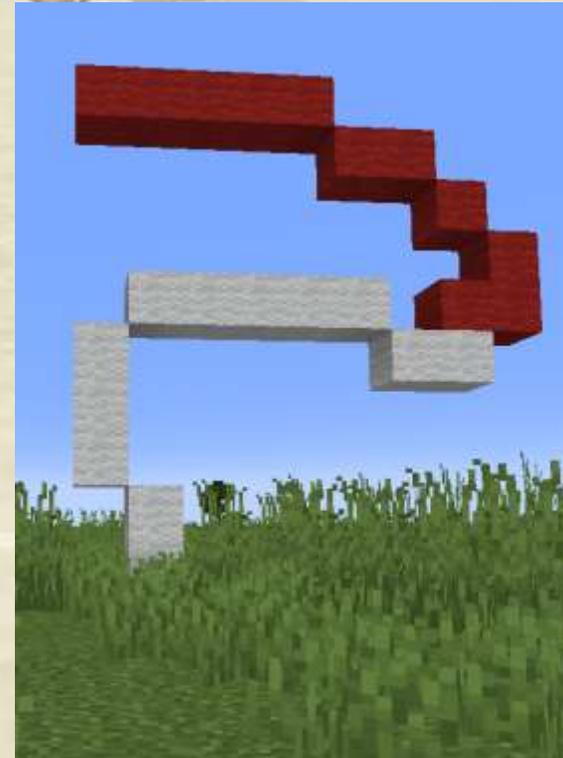
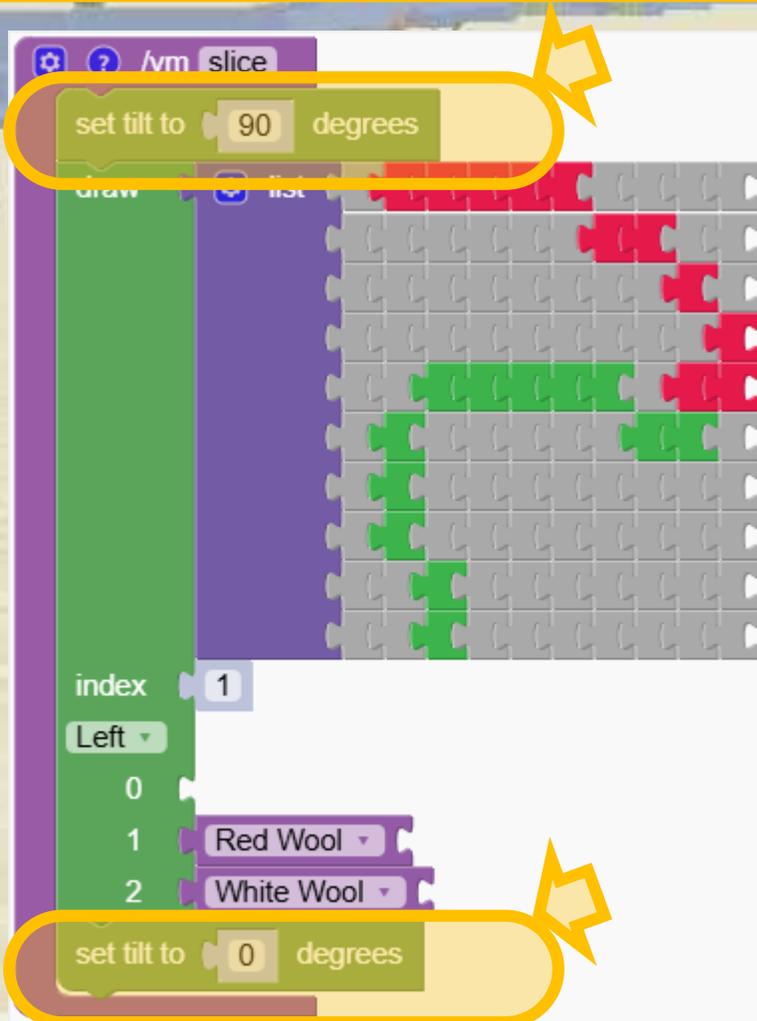
```
draw loop
  list
  index 1
  Left
  0
  1 Red Wool
  2 White Wool
```

The code block is a 'draw' loop with a 'list' block. The 'list' block contains a sequence of wool blocks: a red wool block, a white wool block, and another red wool block. The 'index' block is set to 1, and the 'Left' block is set to 0. The 'Red Wool' and 'White Wool' blocks are connected to the 'list' block.



⚡ The Mushroom House

The slice should be vertical



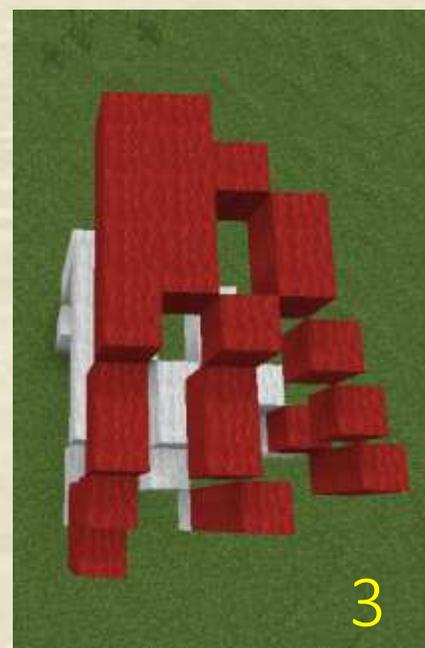
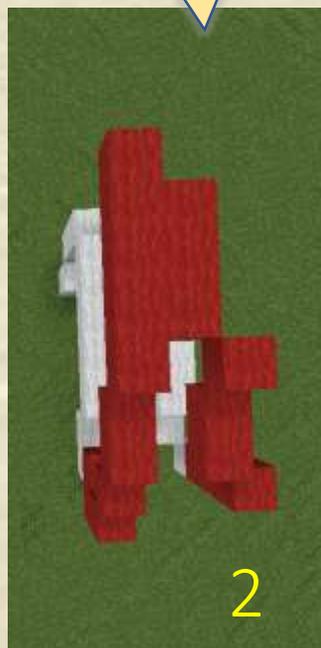
⚡ The Mushroom House

The program below repeats 90 times by rotating the slice

Copied 2 times

Copied 6 times

```
vm house
go 12 blocks forward
repeat 90 times
do slice
  turn right by 4 degrees
```



Conclusion Part 1

What we learned so far

Conclusion

Congratulations on completing the course!

We had a great time exploring Minecraft while learning the basics of coding. From building with commands to automating tasks, we've taken our first steps toward becoming real programmers.

But this is just the beginning! Keep practicing, keep exploring, and don't stop here. Stay tuned for the next parts of the course as they become available—there's so much more to discover!

Happy coding, and see you in the next adventure!

Part 2 (to be published)

Advanced blocks

While loops

Events

Parameters

Part 3 (to be published)

Lists

Return values

Debugging

Coordinate systems