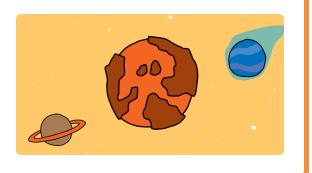


# Create your own world

Create your own open-world adventure game



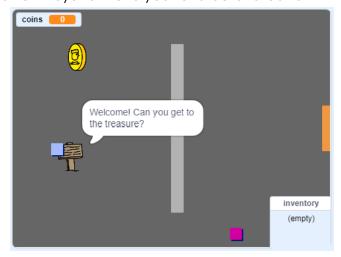


### Step 1 Introduction

In this project, you'll learn how to create your own adventure game world with multiple levels to explore.

#### What you will make

You'll use the arrow keys to move your character around in the world.





### What you will need

#### **Hardware**

• A computer capable of running Scratch 3

#### **Software**

Scratch 3 (either online (<a href="http://rpf.io/scratchon">http://rpf.io/scratchon</a>) or offline (<a href="http://rpf.io/scratchon">http://rpf.io/scratchon</a>)

#### **Downloads**

You can find everything you need to complete this project at rpf.io/p/en/create-your-own-world-go (https://rpf.io/p/en/create-your-own-world-go).



#### What you will learn

- Use conditional selection to react to key presses
- Use variables to store a game's state
- Use conditional selection based on the value of a variable
- Use lists to store data



#### **Additional information for educators**

If you need to print this project, please use the **printer-friendly version** (<a href="https://projects.raspberrypi.org/en/projects/create-your-own-world/print">https://projects.raspberrypi.org/en/projects/create-your-own-world/print</a>).

You can find the solutions to this project at **rpf.io/p/en/create-your-own-world-get** (https://rpf.io/p/en/create-your-own-world-get).

# Step 2 Move the player sprite

Start by creating a player sprite that can move around your world.

Open the 'Create your own world' Scratch starter project.



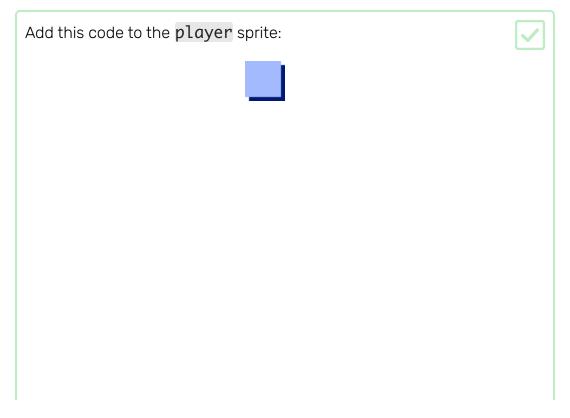
Online: open the online starter project at rpf.io/create-your-own-world-on (http://rpf.io/create-your-own-world-on).

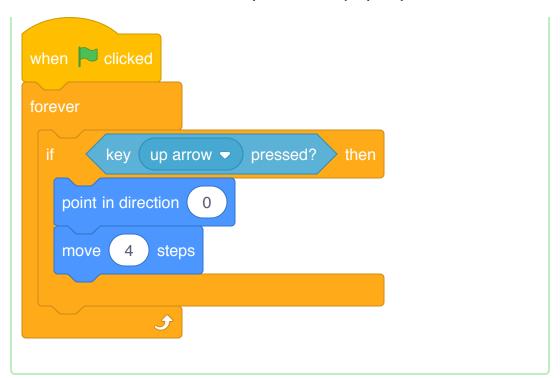
If you have a Scratch account you can make a copy by clicking Remix.

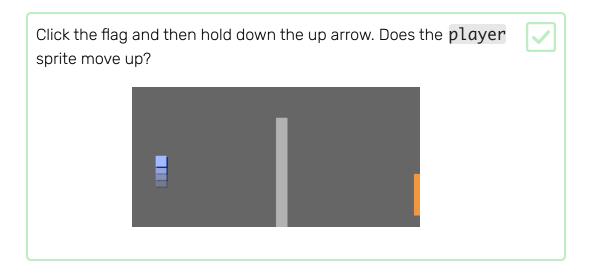
Offline: download the starter project rpf.io/p/en/create-your-own-world-go (http://rpf.io/p/en/create-your-own-world-go), and then open it using the offline editor.

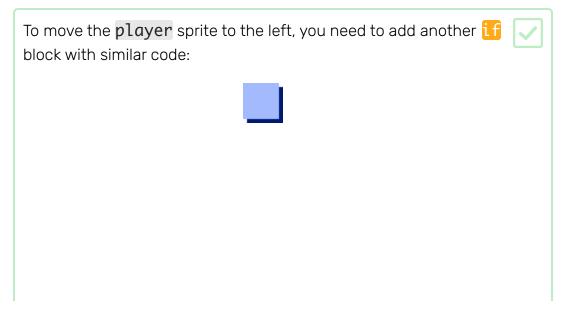
If you need to download and install the Scratch offline editor, you can find it at rpf.io/scratchoff (https://rpf.io/scratchoff).

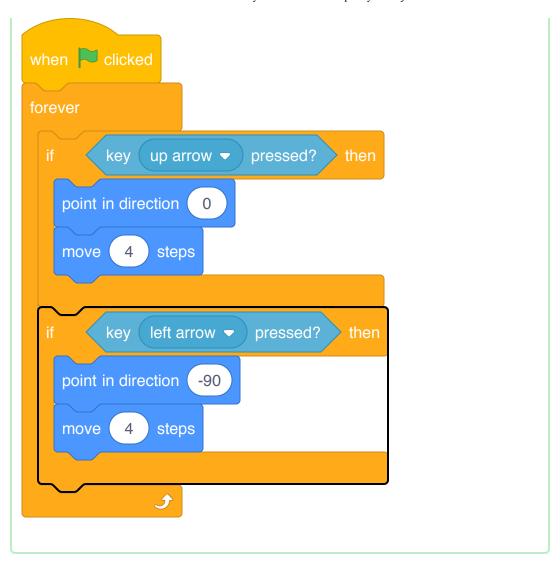
Pressing the arrow keys should move the **player** sprite around. When the up arrow is pressed, the **player** sprite should move upwards on the Stage in response.

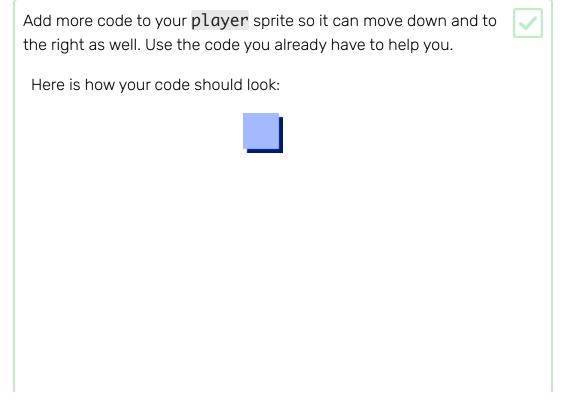


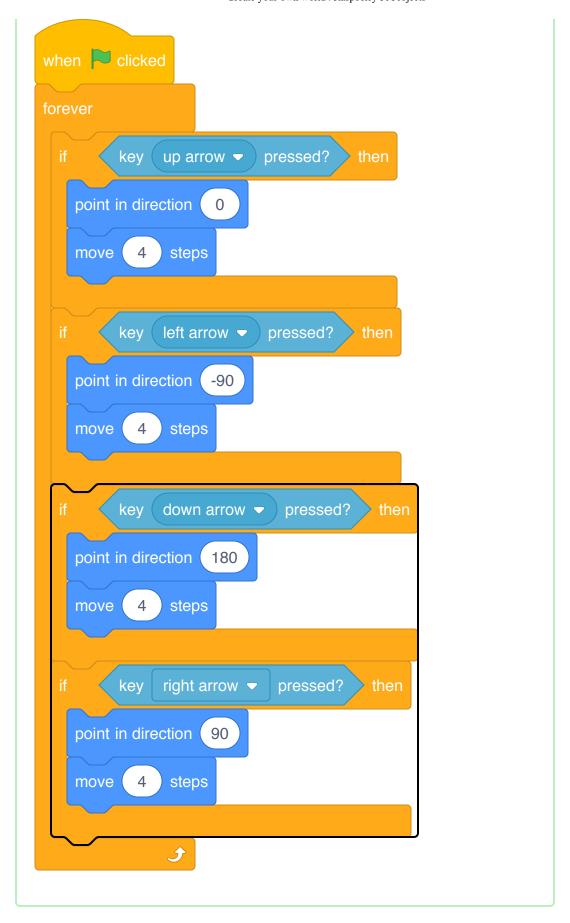




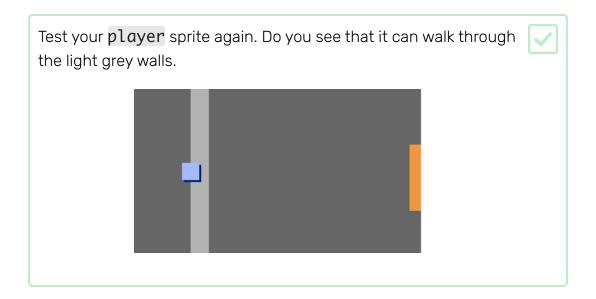


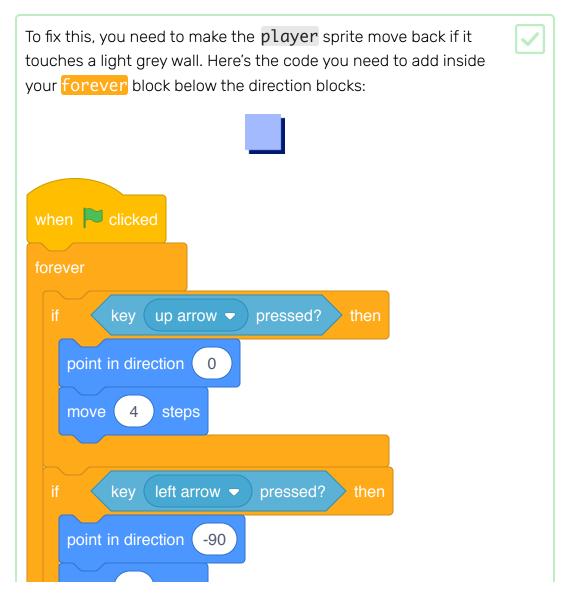


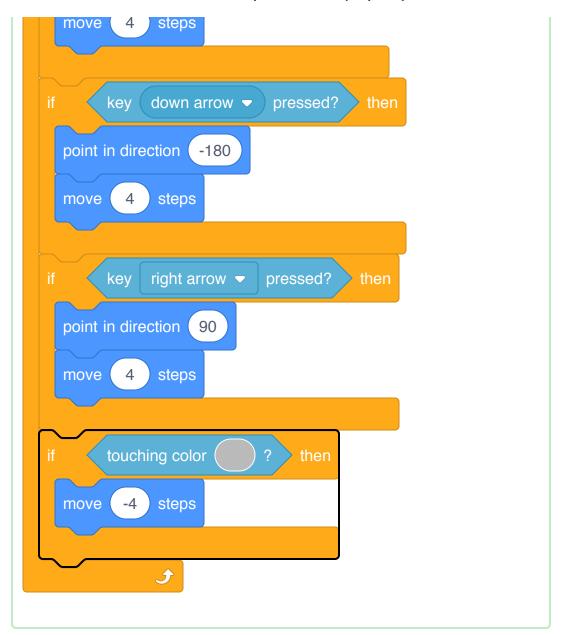


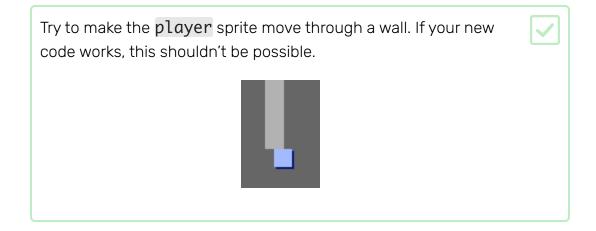


### Step 3 Solid walls







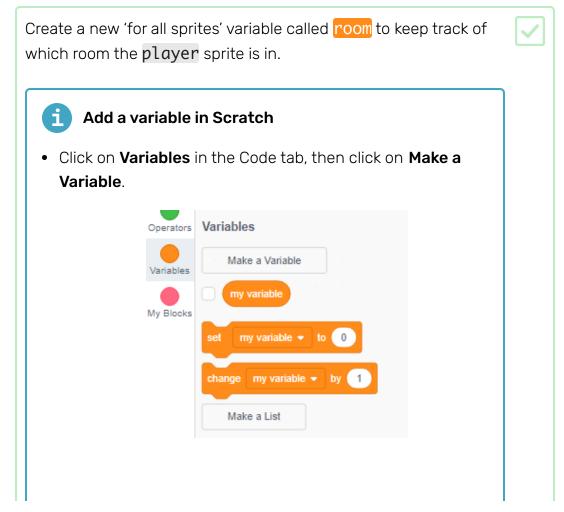


# Step 4 Move around your world

The player sprite should be able to walk through doors into other rooms.

Your project contains backdrops for additional rooms:

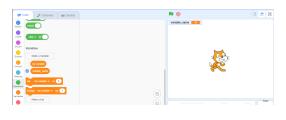




 Type in the name of your variable. You can choose whether you would like your variable to be available to all sprites, or to only this sprite. Press **OK**.



• Once you have created the variable, it will be displayed on the Stage, or you can untick the variable in the Scripts tab to hide it.

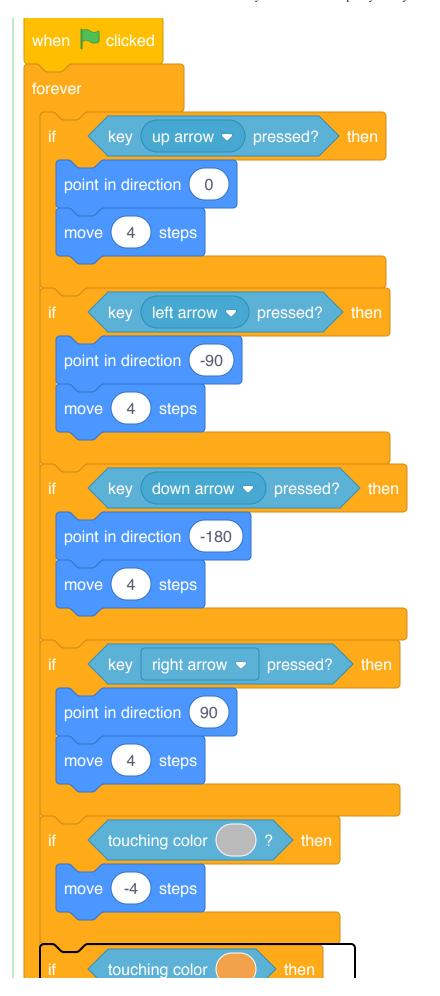


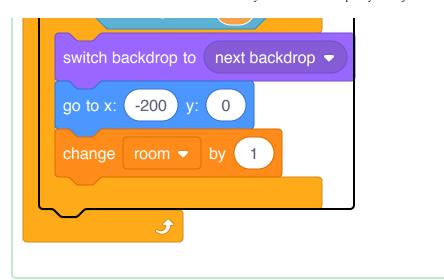


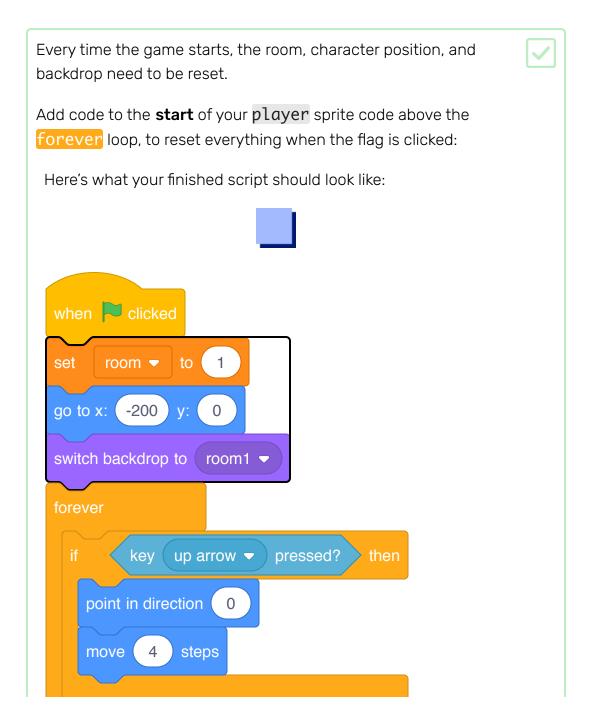
When the **player** sprite touches the orange door in the first room, the game should display the next backdrop, and the **player** sprite should move back to the left side of the Stage. Add this code inside the **player** sprite's **forever** loop:

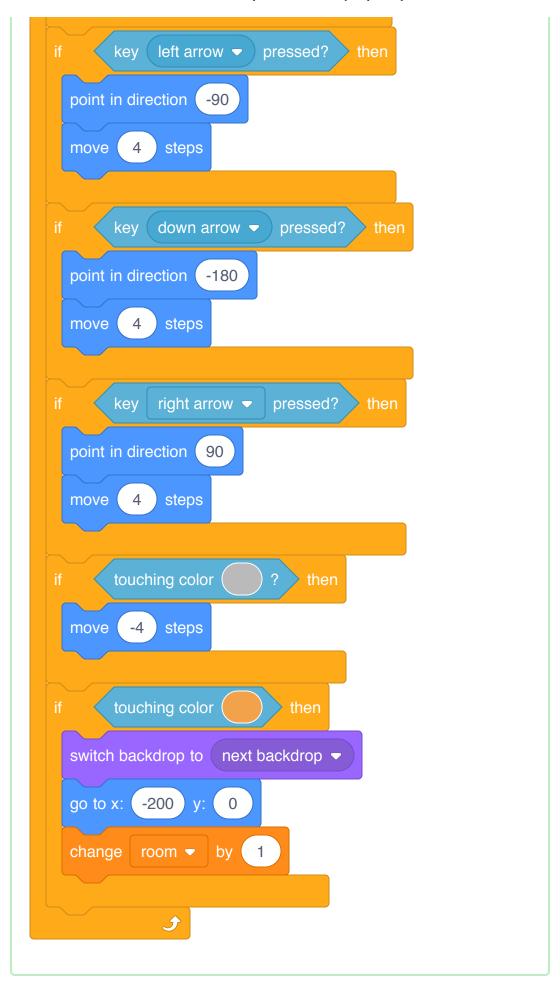












Click the flag, and then move your player sprite until it touches the orange door. Does the sprite move to the next screen? Does the room variable change to 2?



### Challenge: move back to the previous room

Can you make your **player** sprite move back to the previous room when it touches a yellow door? The code you need for this is very similar to the code you've already added for make the sprite move to the next room.

## Step 5 Signs

Now add signs to your world to guide players on their journey.

Your project includes a welcome sign sprite:

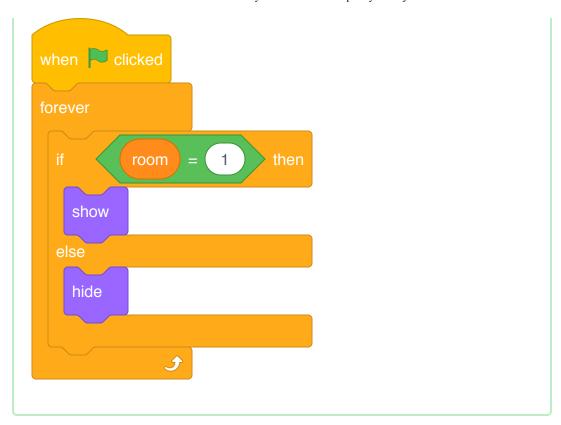


The welcome sign sprite should only be visible in room 1, so add some code to the sprite to make sure that this happens:



Here is the complete code:



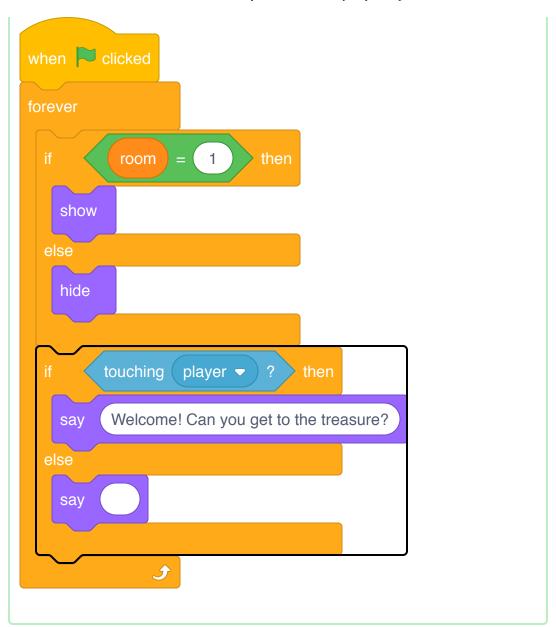




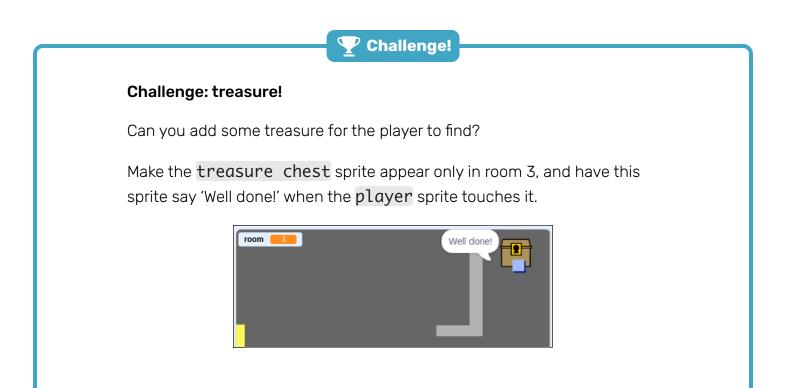
A sign isn't much good if it doesn't say anything! Add some more code to display a message if the **welcome sign** sprite is touching the **player** sprite:





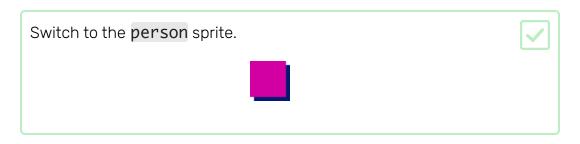






### Step 6 People

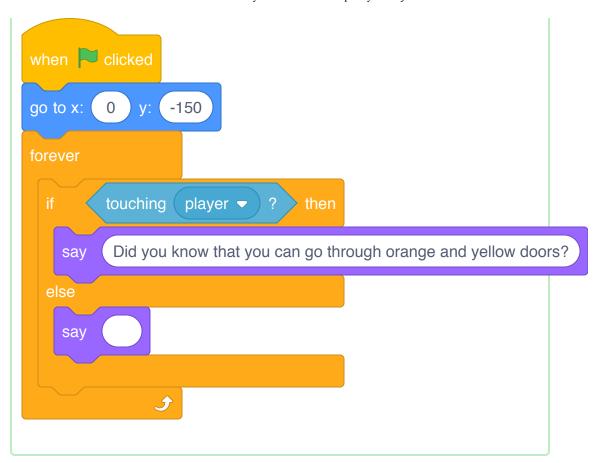
Add other people to your world who your player sprite can interact with.

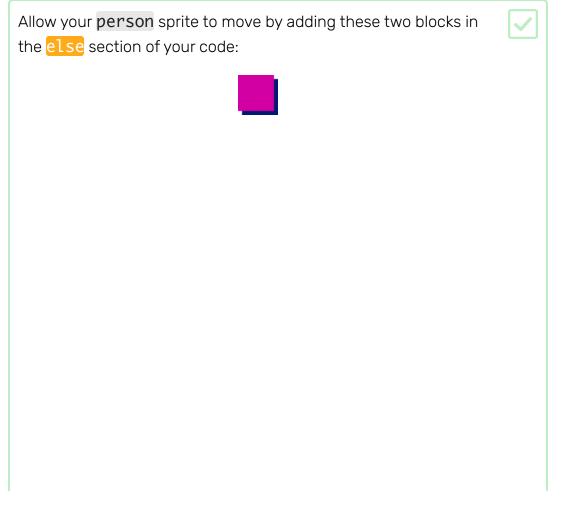


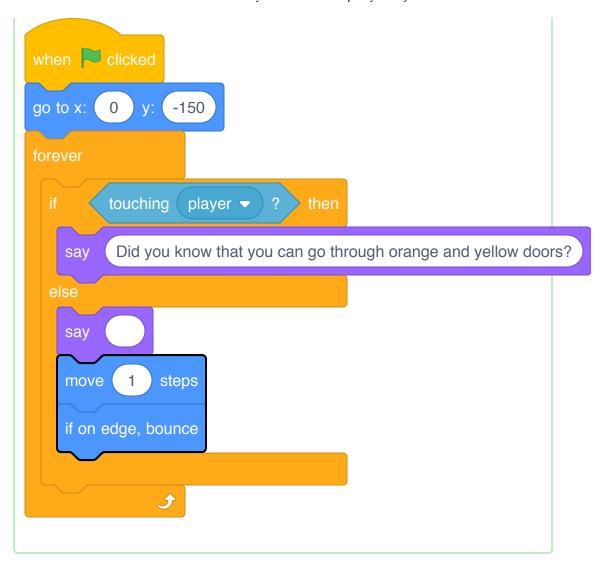
Add some code to the **person** sprite so that the person talks to the **player** sprite. This code is very similar to the code you added to your **sign** sprite:











Your **person** sprite will now move, but will stop to talk to the **player** sprite.



Add code to your new **person** sprite so that the sprite only appears in room 1. The code you need is exactly the same as the code that makes the **sign** sprite only visible in room 1.



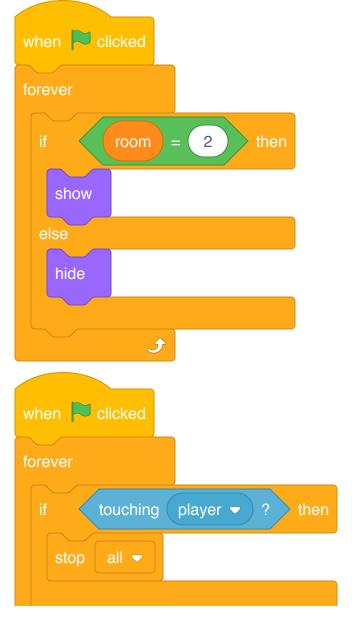
Make sure you test out your new code.

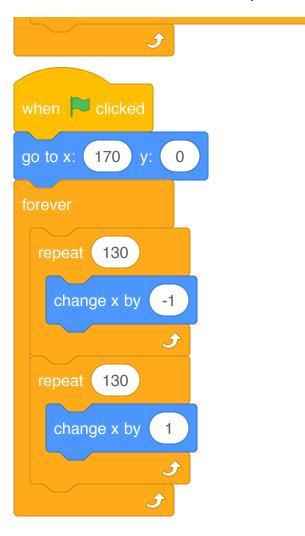


### Challenge: add an enemy

If you want, you can also add patrolling enemies to your game. If the player sprite touches an enemy, the game ends.

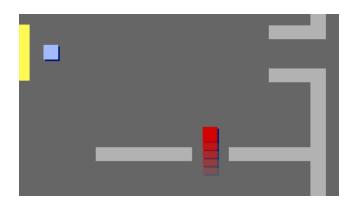
- Your game already contains an **enemy** sprite. Add code to the **enemy** sprite so that it only appears in room 2.
- Add code to move the enemy sprite and to end the game if the enemy sprite touches the player sprite. It's easier to do this in separate code blocks. Here's how your enemy sprite code might look:





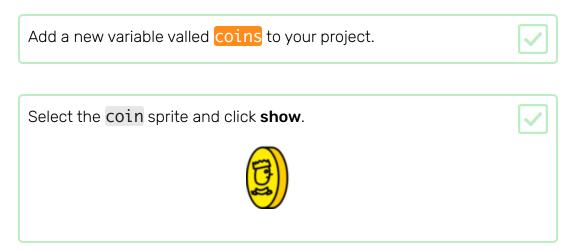
- Test out your new code to make sure that:
  - The enemy sprite only visible in room 2
  - The **enemy** sprite patrols the room
  - The game ends if the player sprite touches the enemy sprite

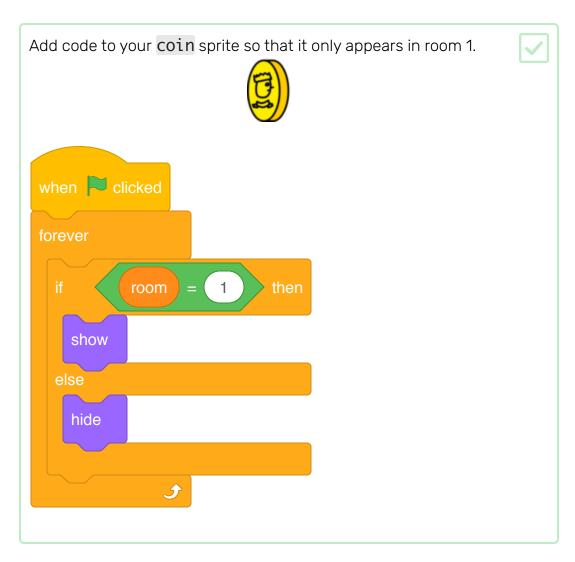
Can you create another **enemy** sprite in room 3 that patrols up and down through the gap in the wall?

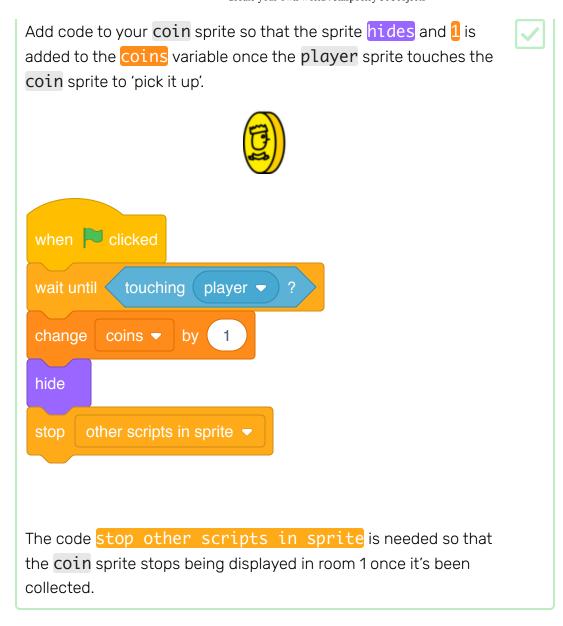


# Step 7 Collect coins

Your **player** sprite should have be able to collect coins as it moves through the world.









Test your game. Collecting a coin should change your **coins** score to 1.



### Step 8 Doors and keys

Now you are going to add code so that some of the doors in your game world are locked, and the player must find the key to open them and get to the next room.

Switch to the **key** sprite. Click on **show** in the Scripts menu so that the sprite appears on the Stage.



Edit the key sprite's costume so that it is blue.



Switch your Stage backdrop to room 3, and place the key sprite somewhere difficult to reach!

Add code to the **key** sprite to make it only visible in room 3.



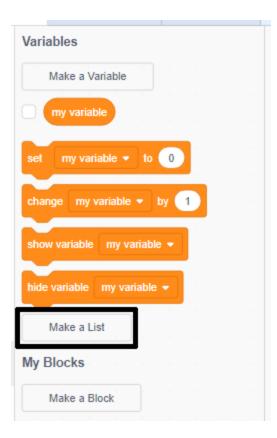
Create a new list called **inventory** to store the items your **player** sprite collects.



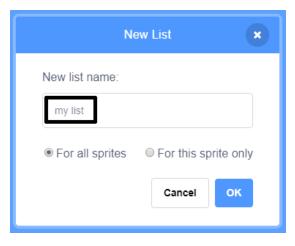


#### Make a list

• Click on Make a List under Variables.



Type in the name of your list. You can choose whether you
would like your list to be available to all sprites, or to only a
specific sprite. Click OK.



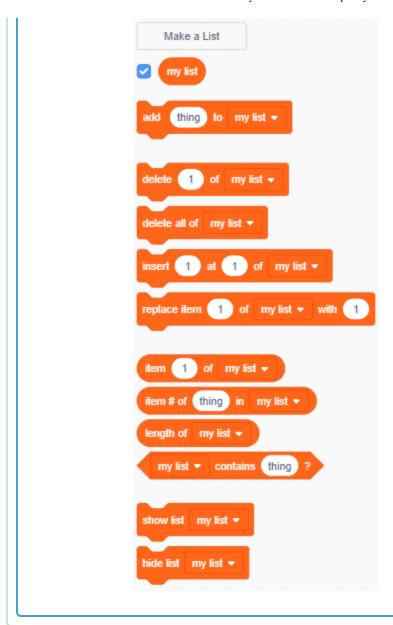
• Once you have created the list, it will be displayed on the stage, or you can untick the list in the Scripts tab to hide it.



• Click the + at the bottom of the list to add items, and click the cross next to an item to delete it.



• New blocks will appear and allow you to use your new list in your project.

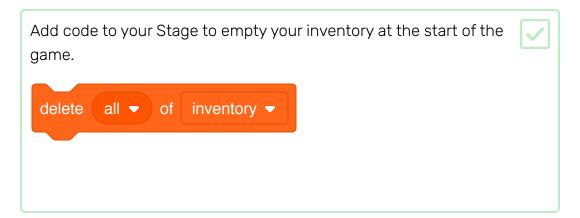


The code you need to add for collecting the key is very similar to the code for collecting coins. The difference is that you add the key to the <u>inventory</u>.

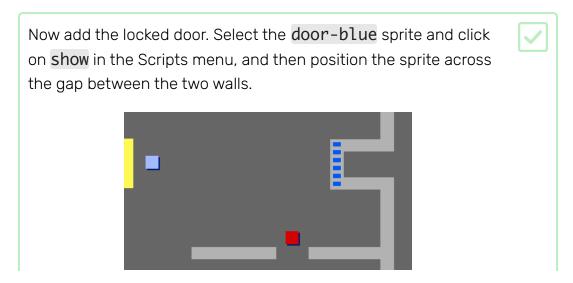






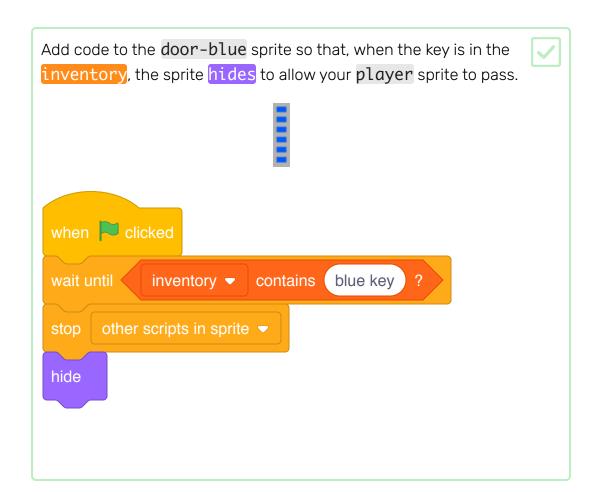


Test out your game to check whether you can collect the **key** sprite and add it to your inventory.



Add code to the **door-blue** sprite so that it is only visible in room 3.





Test out your game and see if you can collect the blue key to open the door!



# **Y** Challenge!

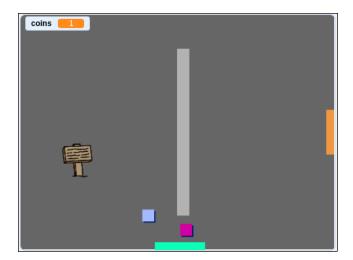
#### Challenge: extend your world

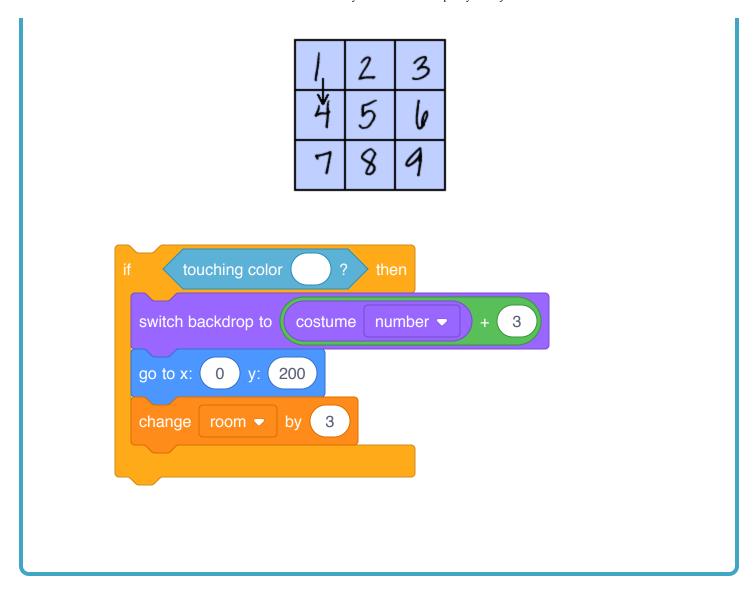
You can now continue creating your own world! Here are some ideas:

- Add more coins to your game in different rooms. Can you let some coins be guarded by patrolling enemies?
- Change your game's backdrops
- Add sound and music to your game
- Add more people, enemies, and signs
- Add red and yellow doors, and special keys to open them
- Add more rooms to your world
- Add other useful items to your game
- Use coins to get information from other people:



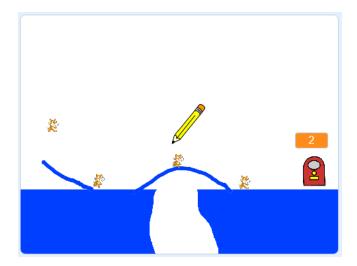
• You could even add doors in the north and south walls of room 1, so that the player can move between rooms in all four directions. For example, your game can have nine rooms in a 3×3 grid. You can then add 3 to the room number to move down one level.





# Step 9 What next?

Have a go at creating another game by working through the **CATS!** (<a href="http://projects.raspberrypi.org/en/projects/cats?utm\_source=pathway">http://projects.raspberrypi.org/en/projects/cats?utm\_source=pathway</a> & utm\_medium=whatnext&utm\_campaign=projects) project.



If you want to make a game using Python instead of Scratch, try out the RPG (https://projects.raspberrypi.org/en/projects/rpg?utm\_source=pathway&utm\_medium=whatnext&utm\_campaign=projects) project.

Published by Raspberry Pi Foundation (<a href="https://www.raspberrypi.org">https://creativecommons.org/licenses/by-sa/4.0/</a>).

View project & license on GitHub (https://github.com/RaspberryPiLearning/create-your-own-world)