

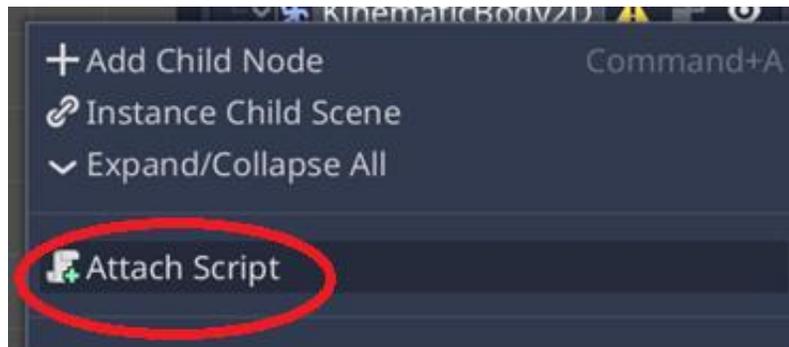
MODULE 2: CREATING THE CODE

Now let's add some code! A script is the code that affects the object.

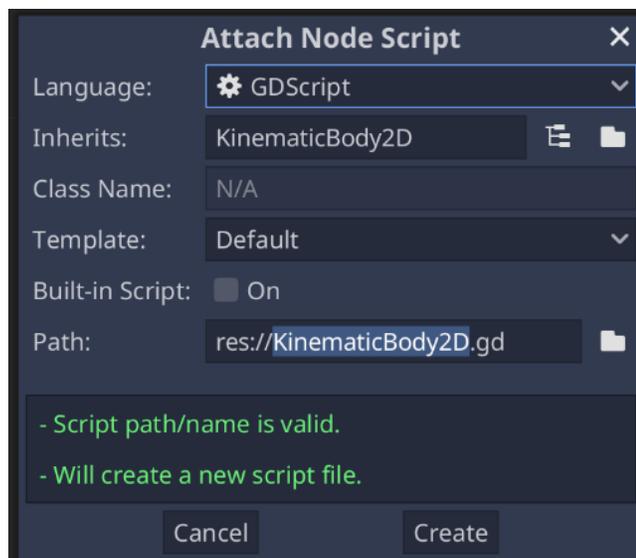
ADD A SCRIPT

First, we attach a script to the KinematicBody2D parent node, and then we write the code.

1. In the Scene widget, right-click "KinematicBody2D".
2. Select "Attach Script".



3. Keep all the settings the same, and click Create.



4. The code script will appear. Delete all the code below line 2. The only code that should remain is line 1: "extends KinematicBody2D".

- Time to write some code! Type all the code as shown below. If you run in to any issues, remember you can always UNDO, via CTRL + Z.

```

1  extends KinematicBody2D
2
3  var motion = Vector2()
4
5  func _physics_process(delta):
6      >|
7      >|     if Input.is_action_pressed("ui_right"):
8          >| >|     motion.x = 100
9          >| >|
10     >|     elif Input.is_action_pressed("ui_left"):
11         >| >|     motion.x=-100
12         >| >|
13     >|     else:
14         >| >|     motion.x=0
15         >|     move_and_slide(motion)
16         >|     pass
17

```

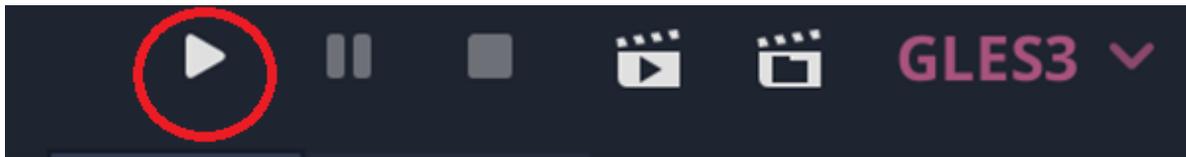
WHAT DOES THIS CODE MEAN?

The code you just wrote tells the software that the object is bound by 2D motion. A physics engine has been put in, and the character can be moved to the left or right in a sliding motion by 100.

TESTING THE CODE

Let's test the game out!

1. Save your game. (Ctrl + S) or in the menu, Scene > Save Scene.
2. Press the play button at the top right of the software, above the scene Widget. (You can also press F5 as a shortcut).
3. Press "Select" to confirm. If it asks you to "Pick a main scene", select your saved file, and open.
4. If there are no errors in the code, the game will open in a new window.
5. Go ahead and play! Using the left and right arrows on your keyboard, your character should move left and right.



PRO TIP:

- If there are any errors, the game will not open correctly. The code must be exact, or it will cause issues.
- If there are errors in the code, they will be highlighted in red, with an explanation on the bottom of how to correct the error(s).
- Correct any error(s) and press the play button. The game should open and work correctly.
- Yellow warnings can be ignored. Yellow warnings state issues with arguments but will not stop the code from running like red errors do.
- Another issue can happen if you choose 3.0 but your computer's graphics does not work at that level. To fix this, you will need to copy the code and recreate the project in 2.0.
- If your new game window is just a grey background (with no player), try maximizing your window.

GRAVITY

Let's add Gravity! We have already taught the game how to move along the X axis (horizontal) in 2D. Now we also need to teach it how to use the Y axis (vertical), so that the character can move up and down.

1. In the Scene widget, go back to your Script under "KinematicBody2D".
2. Add another line of code to line 6: "motion.y+=10".
3. Save the file.
4. Press play.

```
5 v func _physics_process(delta):
6   >|   motion.y+=10
7 v >|   if Input.is_action_pressed("ui_right"):
```

WHAT JUST HAPPENED?

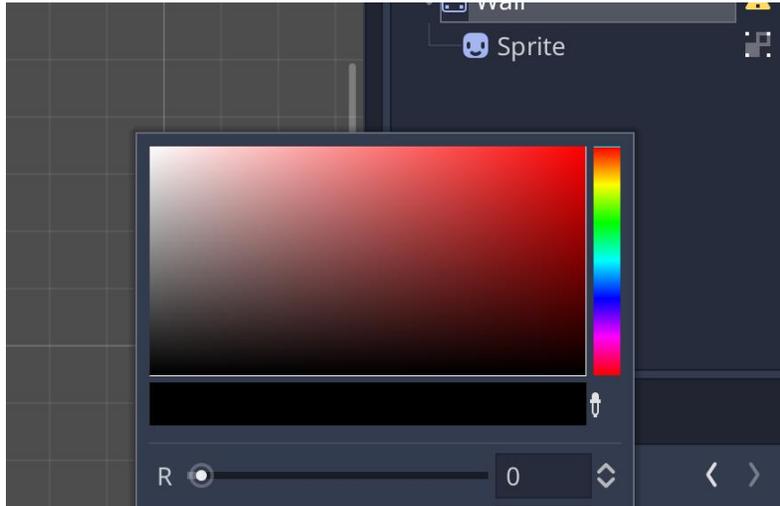
Our character just fell to its doom! Let's create a wall to soften the blow.

1. Add a new StaticBody2D, just like you added the Sprite:
 - a. Go to Scene, click "+".
 - b. Search for "StaticBody2D", and click Create.
 - c. In the Scene widget, double-click StaticBody2D and rename it to "Wall".
2. Add a second Sprite, this time under the StaticBody2D:
 - a. Go to Scene.
 - b. Select "Wall", then click "+" and search for Sprite.
 - c. Once you've created the second Sprite, select it.
 - d. Drag and drop the "icon.png" from FileSystem into the space to the right of "Texture" in the "Inspector" widget (just like when you created the first Sprite in Module 1). You should have two Godot characters on your screen. To turn it into a wall, you'll be changing its colour to black.

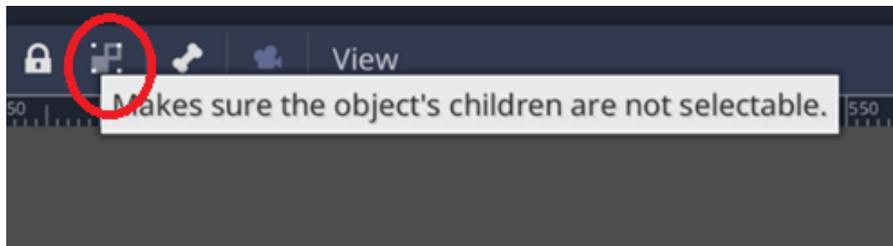
PRO TIP:

- You are selecting "StaticBody2D" rather than "KinematicBody2", because this object will remain static (it will not be moving).

3. Change the colour of the wall:
 - a. Click on “Wall”.
 - b. Go to the Inspector widget.
 - c. Go to Inspector > CanvasItem > Visibility > Modulate.
 - d. Change the colour to Black (hexcode #000000 instead of #FFFFFF).

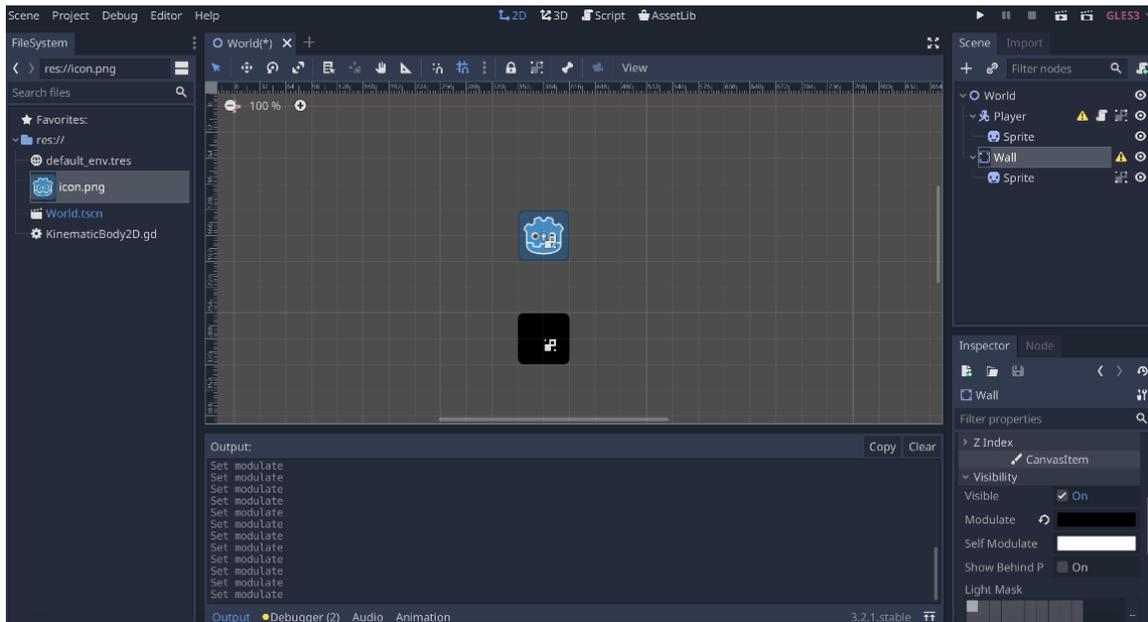


4. Link the Wall and Sprite together.



5. Double-click KinematicBody2D, and rename it to “Player”.

6. Drag the now black wall under the character, and press play.



What happened? Our player just fell through the wall!

[Take me to more Game Development with Godot modules!](#)

[Take me to the Godot Video Tutorial!](#)

[Comments, questions, or feedback? Email us at academy@geeksquad.ca.](mailto:academy@geeksquad.ca)

[Take me back to Geek Squad Academy Online Learning!](#)