# Write your own instruction

In this tutorial, you’ll:

* Practice reading instructions in a script
* Write your own instruction, to declare and assign a new variable
* Test and check your changes to the game

When you’ve finished, you’ll have changed the spawn distance of the potions in the Creator Kit: Beginner Code game.

## **Find the instruction for potion spawn distance**

Now that you’ve explored an instruction which declares a variable and assigns it a value, you’re ready to create your own instruction!

Let’s start by changing the distance at which the potions are spawned from the PotionSpawner GameObject. First, a challenge for you:

* In the SpawnerSample script, find the instruction which tells the computer where the potions should be created. This instruction is repeated three times within the script.

Remember, don’t worry about understanding everything — use variable names, words you recognise, and variable assignment to identify the most appropriate instruction.

## **Break down the instruction**

The instruction which controls the potion spawn distance is repeated on lines **14**, **19**, and **24** of the SpawnerSample script:

spawnPosition = transform.position + direction \* 2;

Let’s break down this instruction:

1. First, work out what you know about the instruction already:

* It includes an equals sign (=), which is used when a value is assigned to a variable. In this case, the variable is called spawnPosition.
* The name of this variable suggests that it contains the potion spawn position. The fact that the instruction is repeated three times (the number of potions which are spawned) supports this.

2. Next, use the [Unity Scripting Reference](https://docs.unity3d.com/ScriptReference/index.html) to search for unfamiliar terms in the instruction. Search for transform.position and select the first result.

On the search result page, you’ll see an explanation that transform.position is the position property of the Transform component attached to a GameObject. This is the same GameObject that spawns the potions: PotionSpawner.

3. The variable named direction stores the coordinates where each potion is spawned.

4. The value 2 is the distance towards that direction where each potion is spawned.

## Recap

This instruction assigns transform.position + direction \* 2 to the variable spawnPosition, to set a different spawn position for each potion bottle.

## **Create your own variable**

So far, you’ve worked out that each potion is created two units of distance away from the PotionSpawner.

This is helpful, but if you wanted to spawn each potion at five units of distance, you would need to change the number in three different places. This isn’t very practical and it leaves room for errors.

To improve this, let’s replace the value with a variable:

1. Write an instruction declaring your own variable called radius to store the distance. Add the instruction on the line following the angle variable (line 10) — this will move all the instructions below it down by one line.

Use the following guidance to help you

* Your variable should be called radius
* The potion should spawn five units of distance away from the PotionSpawner GameObject
* Each instruction needs to end with a semicolon (;)
* The variable should be an int type, so that it can store whole number values

2. Replace the value 2 in the script with the name of your new variable.

## **Check your script changes**

Check your variable declaration against the following solution:

int radius = 5;

Each of the instructions assigning to **spawnPosition** should now match the following example:

spawnPosition = transform.position + direction \* **radius**;

## Save and test your changes

Now you can test your changes at game start:

1. Press Ctrl + S (Windows) or Cmd + S (macOS) in your code editor to save your changes to the script.

2. In the Editor, click Play to enter Play Mode.

3. Check the distance at which the potions are now spawned at game start. You should see that each potion is created further away from the PotionSpawner

4. When you’ve finished checking your changes, click Play again to exit Play Mode. Remember, changes you make in this mode won’t be saved.

Now if you want to change the spawn distance for the potions, you just need to change the value stored in the radius variable!

# Summary

In this tutorial, you have:

* Identified specific instructions in a script
* Applied your own knowledge and used additional resources to work out what an instruction means
* Declared and assigned a new variable
* Replaced a value with a variable in a script

In the next tutorial, you’ll learn how to make things more efficient and avoid some code repetition in your script using functions.

If you’re finding this project challenging, remember that working with code is an ongoing journey. People working on games often use the same skills of deduction and research that you’ve applied in this tutorial to break down the code of more experienced programmers, to enable them to learn and create the best player experience.