# Get started with code in Unity

In this tutorial, you’ll:

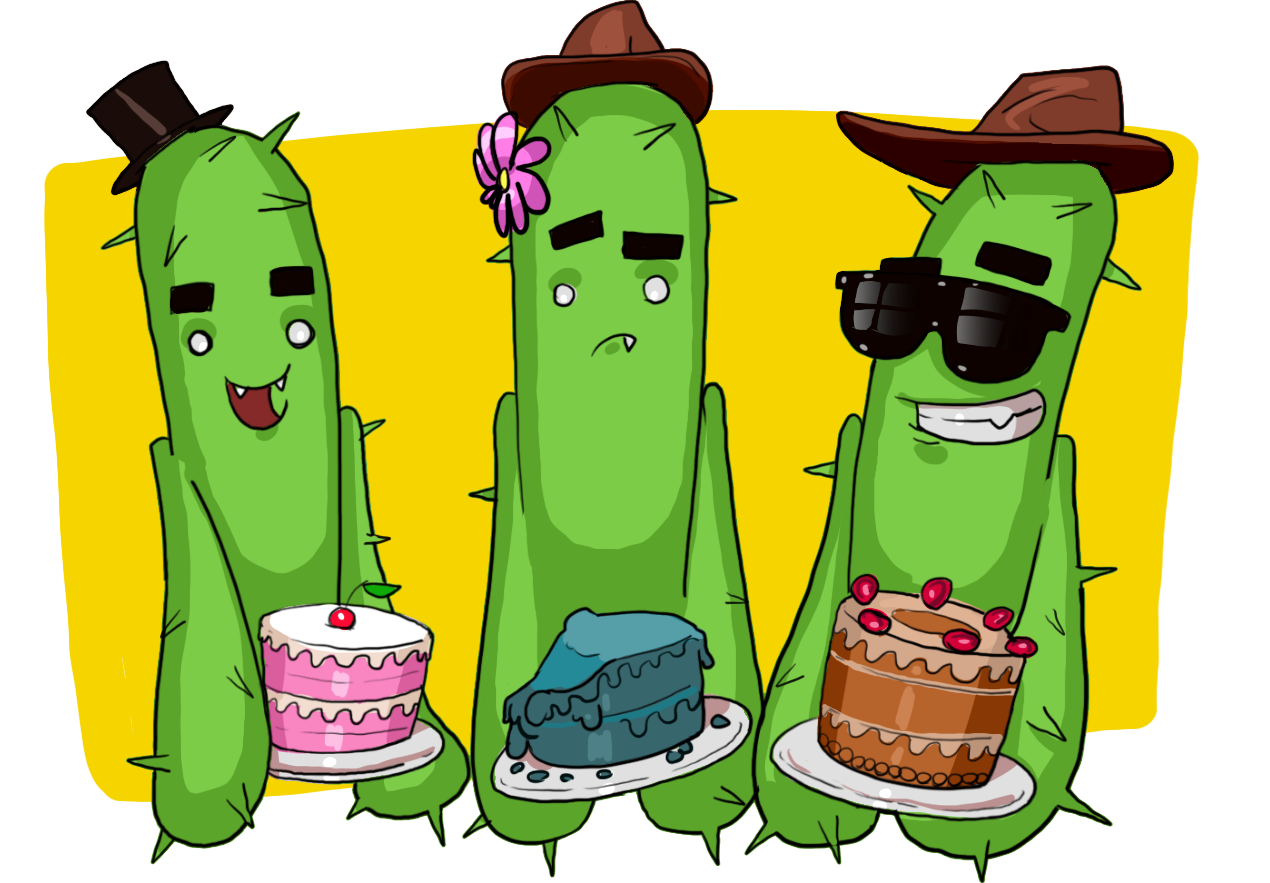
* Learn what programming is
* Explore how scripts work in Unity Editor
* Open a script from the Creator Kit: Beginner Code game in a script editor

When you’ve finished, you’ll be ready to review your first script.

## What is programming?

Before you begin changing the game, let’s review some basic principles for this Creator Kit.

In the simplest sense, programming is giving a computer a list of instructions that it will complete sequentially. This is quite similar to a recipe for a particular food. Just as three bakers might use different recipes to make a cake, three people writing code might approach a problem in a slightly different way.



At the most basic level, computers only understand numbers. This can be challenging for humans; expressing instructions as pure numbers is difficult, if not impossible. To translate between human language and numbers, computers use a type of program called a compiler.

## How do computers understand the instructions?

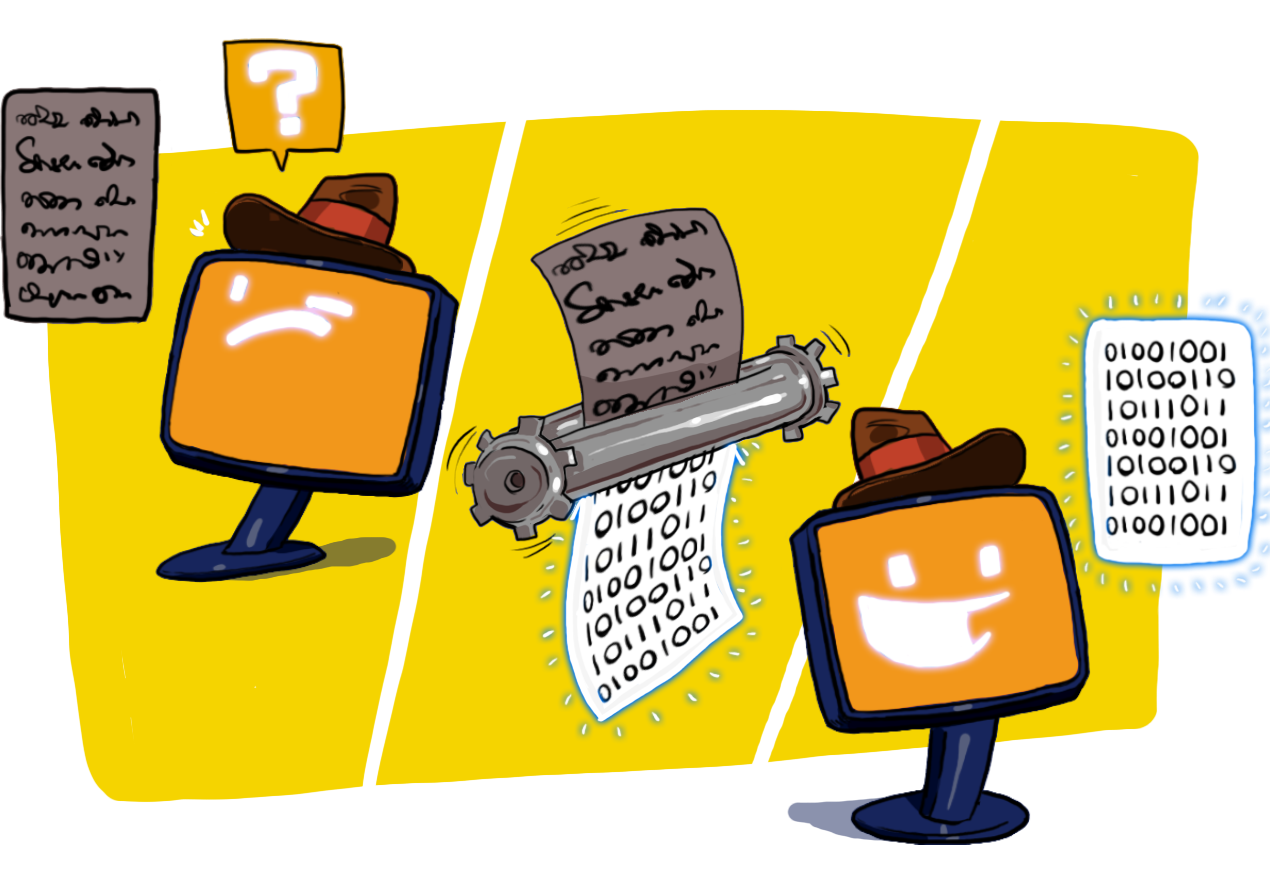
Natural languages are languages which have evolved over time through use (for example, the English language or American Sign Language). It’s hard for computers to understand natural languages — they’re incredibly complex and can be full of nuance. Instead, programmers use specific programming languages.

Programming languages are formal languages, constructed with clear sets of rules. There are a wide range of programming languages currently used for a variety of different purposes.

You need to use the C# programming language to write code for Unity.

## How do I use instructions to create a game in Unity?

When you write C# code for Unity, you need to organize it into documents called scripts. These are sets of instructions telling the computer to do a specific thing. The compiler then turns each script into numbers that the computer can understand, allowing it to run the game as you’ve instructed.



Separating instructions into different scripts is useful because it enables you to:

* Reuse your code in different projects.
* Find and change information quickly when you need to.
* Organize your Project carefully.

This [Creator Kit](https://assetstore.unity.com/packages/templates/tutorials/creator-kit-beginner-code-151986?_ga=2.144000310.200459650.1635526219-1256108084.1635526219) will show you how to find the information that you need in a script, even if you don’t understand everything in it. Its script files are very small, but the same principles apply to the larger scripts you will find elsewhere.

## Open the SpawnerSample script file

GameObjects have components attached. These give specific instructions about their behavior to Unity. There are many built-in components available, which enable you to control different aspects of a GameObject’s behavior. However, to create your own game you will need to use custom scripts and attach them to GameObjects as components.

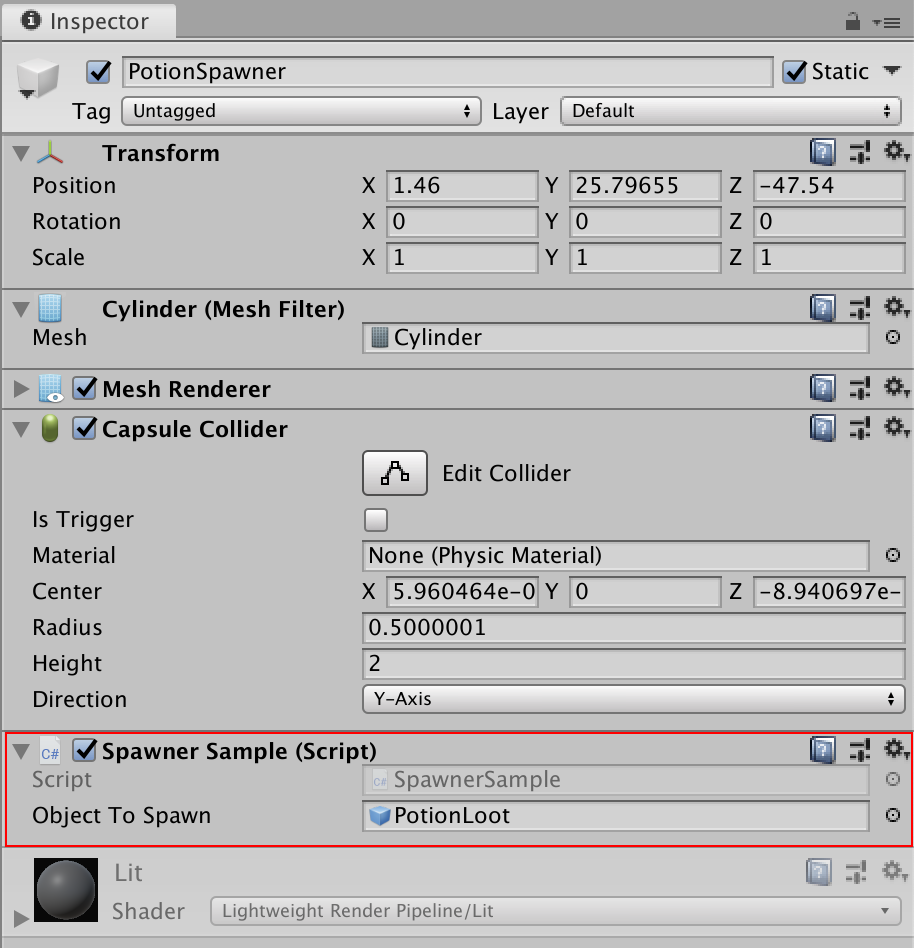
When you tested the game in the previous tutorial, you may have noticed that three health potions appear around the PotionSpawner GameObject at game start.



Let’s start by finding the script file that spawns (or creates) these health potions:

1. In the Hierarchy, select the **LevelDesign > PotionSpawner** GameObject.

2. In the Inspector, find the **Spawner Sample (Script) component**.

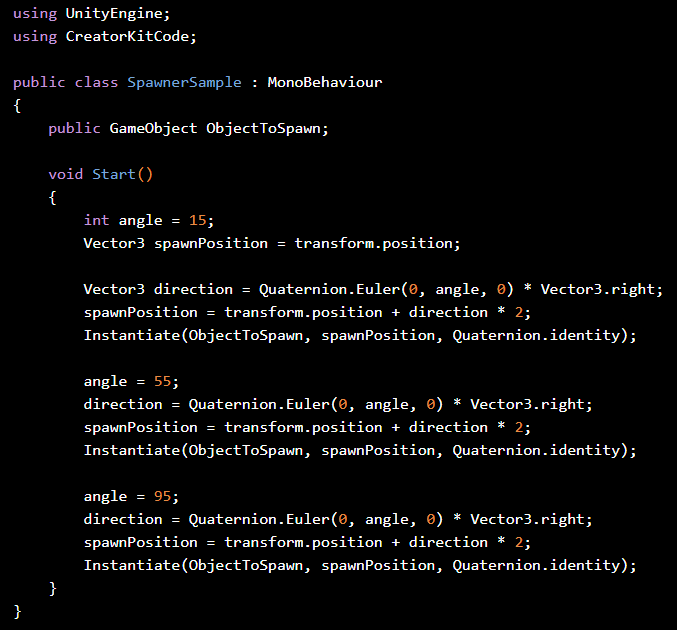


This component is the script which makes the three potions appear.

**3.** In the **Script** field of the component (which will appear greyed out), click on the name of the script (**SpawnerSample**). This will highlight the script file in the Project window.

**4.** Double-click the **SpawnerSample script**. This will open the file in your default installed code editor — unless you have changed the default, this will be Microsoft Visual Studio.

**5.** This is what you will see in the code editor:



If you've never read any code before, this may look overwhelming ⁠— don’t panic! In the next tutorial, you’ll learn to identify basic code elements in a script file even if you don’t understand all of its content.

## Summary

In this tutorial, you have:

* Reviewed the fundamentals of programming
* Learned that scripts are a series of instructions for the computer written in code
* Found a specific script in Unity Editor
* Opened the script in your computer’s default code editor

In the next tutorial, you’ll explore your first C# script and learn about a key ingredient for writing efficient code: variables.