



# MINECRAFT

## EDUCATION EDITION

### Educator Guide

Minecraft Python Course - Lesson 2

45 minutes

---

## Location, location, location...

**PARAMETERS AND COORDINATES**

[EDUCATION.MINECRAFT.NET](https://education.minecraft.net)

## LESSON OBJECTIVES:

In this lesson the students will learn:

- the concept of relative coordinates.
- positional programming in Minecraft.
- command in command structure.
- to use the code completion tool.

## LESSON INTRODUCTION: 10 minutes

### Theme:

Tell the students that they need to help Coding Mine with their new project to expand their office space. They are working with engineers to make this happen, however, CodingMine is also developing software to make the construction process easier. This software will automatically place blocks in the desired locations. The students need to write down Minecraft Python commands to place blocks in the right locations in a three-dimensional space and find the most efficient way to construct objects.

### Coding Concepts:

#### Commands inside commands

Tell the students that they will be combining commands one inside the other. This means that they will be using certain commands as parameters inside other commands. Not all commands can be used as parameters; mostly only commands that return some kind of text or mathematical value are used inside other commands. An example of such a command structure would be:

```
blocks.place(GRASS, pos(0,0,0))
```

#### Relative position

Tell the students there are two kinds of positional coordinates in Minecraft, relative position and world position. They will be working with relative position in this lesson. A relative position is defined by three numbers that locate a position relative to the player in Minecraft. An example of this would



be the position `pos(0, 0, 0)`. This is the position of a block at the player's feet in Minecraft.

If we change the first number (the first zero) to 1 this would mean one block east of the player, if the number was -1 it would mean one block west of the player.

The second number defines the number of blocks up or down. If the number was two, the block would be above the player's head, as the Minecraft character takes up two blocks `pos(0, 0, 0)` and `pos(0, 1, 0)`. If the number was -1 the block would be below the characters feet.

If we change the third number to 1 this would mean one block south of the player, if the number was -1 this would mean one block north of the player.

None of these numbers are written with quotes as they all represent a numerical value. During the lesson the students will be able to see north, south, east and west directions on the walls to help them determine which direction to build in.

### User Interface:

In this lesson the students will:

- use commands by dragging and dropping them into the coding window.
- edit commands placed in the coding window.
- use the code completion tool to select correct block names.

### Syntax / Operators:

Tell the students in this lesson they will learn about:

#### Commas , :

Commas are used when programming commands to separate parameters if there are two or more. An example of this would be:

```
command_group.command_name(parameter1, parameter2, parameter3)
```



## **CODING ACTIVITIES:** 30 minutes

**Activity 1:** Place the correct block.

**Objective:** Explain to the students that the programmer needs their help to develop software that is going to place specific block types on the marked areas. They will be doing this by using the `place block at position` command from the `block` tab. In this command they will be changing the first parameter (block) to place the desired block type. They do not need to change the second parameter, position of the block; it should be on its default setting (0, 0, 0).

Tell the students to place two blocks in the marked area. *(Hint: If a block is placed incorrectly the students can reset the Activity by pressing the try again button when talking to the NPC. This is true throughout the course).* When the students have placed both blocks, Activity one is complete and the door to the next Activity will open.

Code snippets:

```
blocks.place(GRASS, pos(0, 0, 0))
```

```
blocks.place(BRICKS, pos(0, 0, 0))
```

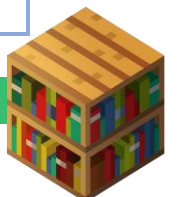
**Activity 2:** All of the lights.

**Objective:** Explain to the students that the electrical engineer needs their help to place ceiling lights at different heights. They will be doing this by changing the second coordinate of the second parameter in the `place block at position` command. Tell the students that to determine the required heights more easily, measuring scales have been placed next to each of the designated areas and they should refer to these when coding.

When the student has placed the lamps in the fittings tell them to use the switches to turn on all the lights. When all the ceiling lights are lit. Activity 2 is complete and the door to the next Activity will open.

Code snippet:

```
blocks.place(REDSTONE_LAMP, pos(0, 6, 0))
```



```
blocks.place(REDSTONE_LAMP, pos(0, 7, 0))
```

```
blocks.place(REDSTONE_LAMP, pos(0, 4, 0))
```

**Activity 3:** One step at a time.

**Objective:** Explain to the students that the construction engineer wants their help to complete a set of stairs. The students will be constructing these stairs by using the second and third coordinate of the `place block at position` command.

Tell the students to start by standing on the gold and then to look at the walls where it shows the north, south, east and west directions. Tell them that they have to place the blocks in the empty spaces in front of them. When they have finished building the set of stairs, the third Activity is complete, and the students can climb the stairs to the next Activity.

**Code snippet:**

```
blocks.place(BRICKS, pos(0, 0, 1))  
blocks.place(BRICKS, pos(0, 1, 2))  
blocks.place(BRICKS, pos(0, 2, 3))
```

**Activity 4:** Getting across.

**Objective:** Explain to the students that the next construction engineer wants their help to complete the floor of the hallway. They will be constructing this floor by using negative numbers in the first and second coordinate of the `place block at position` command.

Tell the students to start by standing on the gold on the floor, and then to look at the walls where it shows the directions. Tell them that they need to place five blocks one, in front of the other. When they have finished the line of floor, the fourth Activity and lesson is complete, and the students can cross to the other side to reach the lift.

**Code snippet:**



```
blocks.place(PLANKS_OAK, pos(-1, -1, 0))
blocks.place(PLANKS_OAK, pos(-2, -1, 0))
blocks.place(PLANKS_OAK, pos(-3, -1, 0))
blocks.place(PLANKS_OAK, pos(-4, -1, 0))
blocks.place(PLANKS_OAK, pos(-5, -1, 0))
```

## LESSON CONCLUSION: 5 minutes

Ask the students about the skills that they have learned during the lesson, to reinforce the concepts.

1. Q. What command do we use if we want to place a block in a relative position?  
A. The `place block at position` command
2. Q. What is a **relative position**, relative to?  
A. It is relative to the player's position in Minecraft.
3. Q. Are coordinates written in quotes?  
A. No, because coordinates are a numerical value.
4. Q. Look at the code: `blocks.place( BRICKS, pos(0, -1, 0) )`.  
Where is the bricks block placed?



## EDUCATION STANDARDS:

CSTA K-12	
1A-AP-11	Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.
1B-AP-08	Compare and refine multiple algorithms for the same task and determine which is the most appropriate.
2-AP-17	Systematically test and refine programs using a range of test cases.
ISTE	
7A	Provide alternative ways for students to demonstrate competency and reflect on their learning using technology.
3B	Establish a learning culture that promotes curiosity and critical examination of online resources and fosters digital literacy and media fluency.
6B	Manage the use of technology and student learning strategies in digital platforms, virtual environments, hands-on makerspaces or in the field.



## COMMANDS:

### Relative position

```
pos(0, 0, 0)
```

~ **x** ~ **y** ~ **z**

Description: Creates a new relative position.

x: The east (+x) or west (-x) coordinate, in blocks.

y: The up (+y) or down (-y) coordinate, in blocks.

z: The south (+z) or north (-z) coordinate, in blocks.

### Place block at position

```
blocks.place(block, pos(0, 0, 0))
```

place **block** at **pos**

Description: Places a block at a certain position.

block: Defines what block type should be placed.

pos: Defines the position where the block is placed.

This command structure is a combination of two different commands, `blocks.place()` and `pos(0, 0, 0)`.

