MINECRAFT
EDUCATION EDITION

Educator Guide

Minecraft Python Course – Lesson 3
45 minutes

A Varied Outcome

VARIABLES

EDUCATION.MINECRAFT.NET

LESSON OBJECTIVES:

In this lesson the students will learn:

- to create a variable and assign a string to it.
- to create a variable and assign a numerical value to it.
- to change the numerical value of a variable.
- the concept of world coordinates.
- to use predefined locations.

LESSON INTRODUCTION: 10 minutes

Theme:

Tell the students that they need to help CodingMine with the development of their new farming software. This software will be used to help farmers in every step of the farming process from planting and sorting to selling the produce. Explain to the students that they will be using variables in this lesson and world coordinates.

Coding Concepts:

Variables

Tell the students that they will learn about variables. Variables are used as placeholders for pieces of text (strings), numerical values, commands, and even other variables. Tell the students that a variable inside a piece of code is replaced by a number or text when the code is run. Tell them that you can think of a variable as a box that contains something (a number or text). For example, if you wanted to make a box/variable for countries, you could name your box/variable **country**. You can then pass the box called **country** to another student who places a piece of paper with the name of a country (**America**) inside the box. This gives the box/variable **country** a value of **America**. The box is then moved to another student who places their country of choice (**France**) inside the box and removes the previous one (**America**). The box/variable now has the value (**France**). Any time you want to see the current country, you can open the box/variable **country** and have a look.

World position

Tell the students there are two kinds of positional coordinates in Minecraft, **relative position**, and **world position**. In this lesson they will be working with

world position. Every block in the Minecraft world has its own unique coordinates called **world position**, defined by three numbers `world(0, 0, 0)`.

At the start of Activity 2 the world coordinates will appear in the top left-hand corner. Coordinates in commands are not written with quotes as they represent a numerical value. During the lesson the students will be using predefined coordinates in their code and do not need to change any of their values.

User Interface:

In this lesson the students will:

- see displayed world coordinates in the top left-hand corner.
- learn to use the half coding window, giving the students the possibility to see the coding window and world position at the same time.
- use predefined given world locations, at the top of the coding window.

Syntax / Operators:

Tell the students in this lesson they will learn about:

Equals = :
Equals is used with variables to declare its value.

Change += :
Increases the value by the defined amount.

Change -= :
Decreases the value by the defined amount.

CODING ACTIVITIES: 30 minutes

Activity 1: Sorting things out.

Objective: Explain to the students that the programmer needs their help to develop software that is going to sort the different fruit that the farmers are picking. The students will need to work out the correct sequence of buttons to press on the sorting machine. They will be doing this by creating variables for the different fruits and finishing off the code so that when it is run the correct sequence is shown.

Tell the students to run the code and press the buttons for each fruit in the order that it is shown in the chat. When the students have entered the correct combination, the lights will be fully lit, and Activity 1 is complete.

Code snippet:

Before:

```
# Replace the lines below with your code #
a = "berries"
# variable b | Step 1
# variable c | Step 1
d = a
# fruit variable      | Step 2
player.say(fruit)

# fruit variable      | Step 2
player.say(fruit)

# fruit variable      | Step 2
player.say(fruit)

fruit = d
player.say(fruit)
```

After:

```
a = "berries"
b = "melon"
c = "apple"
d = a

fruit = a
player.say(fruit)

fruit = c
player.say(fruit)

fruit = b
player.say(fruit)

fruit = d
player.say(fruit)
```

Activity 2: Ideal planting spot.

Objective: Explain to the students that the developer needs their help to program a piece of code that will plant the correct crop in a desired location. The students need to do this by matching the world positions in the Minecraft world with the different location variables in the coding window. This is done by looking at the world position in the top left-hand corner when the player is standing in the right location. Once the students have found the location, they should place a block, using code, that corresponds to the type of block hanging above the location (melon block or pumpkin) using the place block at position command with that location variable as the position. Tell the student to use the half coding window button in the top left-hand corner of the coding window to be able to see the world position and coding window at the same time.

(Hint: The students will be given predefined locations at the top of the coding window, called location1, location2, location3... They should use these

locations when coding). When the students place the last crop, Activity 2 is completed.

Code snippet:
Before:

```
location1 = world(-24, 40, -18)
location2 = world(-31, 40, -11)
location3 = world(-28, 40, -16)
location4 = world(-25, 40, -13)
location5 = world(-31, 40, -17)
# Replace the lines below with your code #

# place block at location1 command
# place block at location2 command
# place block at location3 command
# place block at location4 command
# place block at location5 command
```

After:

```
location1 = world(-24, 40, -18)
location2 = world(-31, 40, -11)
location3 = world(-28, 40, -16)
location4 = world(-25, 40, -13)
location5 = world(-31, 40, -17)

blocks.place(MELON_BLOCK, location1)
blocks.place(MELON_BLOCK, location2)
blocks.place(PUMPKIN, location3)
blocks.place(PUMPKIN, location4)
blocks.place(MELON_BLOCK, location5)
```

Activity 3: Are my calculations off?

Objective: Explain to the students that the data scientist needs their help to write three pieces of code that will calculate the cost of the fruit when the farmers are selling them. They will have to code three different calculations for different situations. Examples would be when the price of fruit increases or decreases and when the farmers add a new type of crop. Once calculated they will have to show the total cost in the chat using the say command. The students should then press the button that corresponds to the cost shown in

the chat. Tell the students that they will be working on one piece of code in this Activity, adding additional parts as needed.

Whenever the students run the code, the say command will show the numerical value of the variable `cost` in the chat. Tell the students to press the button that corresponds to that number and if correct, a receipt will print out. When the students have pressed all of the last correct button, they will receive their last receipt. The students will then have completed the Activity 3 and the lesson.

Code snippets:

Before:

```
apple = 10
melon = 15
berries = 20
potato = 2
# Replace the lines below with your code #
# replace with pumpkin variable | Step 2
# apple change                            | Step 3
# melon change                            | Step 3
# cost variable        | Step 1
player.say(cost)
```

After:

```
apple = 10
melon = 15
berries = 20
potato = 2
# Replace the lines below with your code #
# replace with pumpkin variable | Step 2
# apple change                            | Step 3
# melon change                            | Step 3
cost = apple + melon + (berries * 2) + potato
player.say(cost)
```

Hint: Total cost = 67

```
apple = 10
melon = 15
berries = 20
potato = 2
pumpkin = berries - apple
# apple change                              | Step 3
# melon change                              | Step 3
cost = apple + melon + (berries * 2) + pumpkin + potato
player.say(cost)
```

Hint: Total cost = 77

```
apple = 10
melon = 15
berries = 20
potato = 2
pumpkin = berries - apple
apple += 2
melon -= 3
cost = apple + melon + (berries * 2) + pumpkin + potato
player.say(cost)
```

Hint: Total cost = 76

LESSON CONCLUSION: 5 minutes

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

1. Q. What is a variable?
   A. A variable is a box/placeholder for a numerical value, piece of text, command, or another variable.
2. Q. How do you set the value of a variable?
   A. Using equals, e.g. `a = 1`.
3 Q. If **apple** = 3 and we use **apple+=2**, what is the new value of apple?
   A. 5.

4. Q. Each block in the world has its own unique coordinates. What is this called?

   A World position.

EDUCATION STANDARDS:

CSTA K-12	
1A-AP-11	Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.
1A-AP-08	Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.
2-AP-17	Systematically test and refine programs using a range of test cases.
1B-AP-09	Create programs that use variables to store and modify data.
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:

World position

```
world(0, 0, 0)
```

Description: Creates a new world 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))
```

Description: Places a block at a certain position.

block: Defines what block type that should be placed.

pos: Defines the position where a block is placed.

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

Say

```
player.say(message)
```

Description: Shows something in the chat, in the game.

message: The message that the player wants to display in the chat. This message can be either a piece of text (string) or a mathematical value.

Equals

```
=
```

Description: Sets the value of a variable.

Change

```
item += 1
item -= 1
```

change

Description: Changes the value of a variable.