



# MINECRAFT

## EDUCATION EDITION

### Educator Guide

Python Island 2

45 minutes

Single Student

**Using conditional logic (if statements) in Python.**

**THEME OVERVIEW**

**Causeway Digital**

Welcome to the Kingdom of the Floating Islands. This mystical kingdom floats high above the world below. Throughout this collection of lessons, students will learn the basics of Python, through a series of themed tasks in each world. Completing these tasks will allow the kingdom to develop further technologically.

## LESSON OBJECTIVES

- Understand that computers make decisions all the time.
- Become familiar with the concept of an “if” statement, to make these decisions in Python.
- Build up experience with multiple different approaches using conditional statements
  - Comparing Booleans.
  - Comparing Integers.
  - Using “else” statements.
- Continue to develop an understanding of the concept of **Decomposition**, as students break problems down into codable solutions for the Agent.
- Continue to see the importance of **Sequencing**, putting things in order, so that the Agent can complete its tasks.

## THINGS TO KEEP IN MIND

- Students are given a whistle in the first slot of their hotbar. This allows students to teleport their Agent to them at any point.
- Remind students there may be more than one solution for each of the activities.

## MINECRAFT MECHANICS

C	C Summons the Agent and opens the Notebooks interface.
T	T Opens chat panel in Minecraft for commands to be typed
ESC	ESC When a student wants to leave the game, leave chat, or pause the game.

## PYTHON COMMANDS

<code>say ("Message")</code>	Say command Output a message in chat
------------------------------	---



<code>agent.move("Direction")</code>	<b>Agent move</b> Tells the Agent to move in a certain direction 1 block.
<code>agent.place(1, "Direction")</code>	<b>Agent place</b> Tells the Agent place a block from inventory slot 1, in a certain direction.
<code>agent.destroy("Direction")</code>	<b>Agent destroy</b> Tells the agent to destroy/break a block in a certain direction.
<code>agent.till("Direction")</code>	<b>Agent till</b> Tells the agent to till a block of dirt ready for planting seeds, in a certain direction.
<code>agent.inspect("forward")</code>	<b>Agent inspect</b> Get the name of the block in the direction the agent is checking.
<code>if (logical expression):     statement() else:     statement()</code>	<b>If / else statements</b> Compare 2 values. If they are equal, run one piece of code. If they are not equal, run a different piece of code (else).

## KEYWORDS

**Conditional Statements** – Decision statements, aka “if” “else” statements in Python.

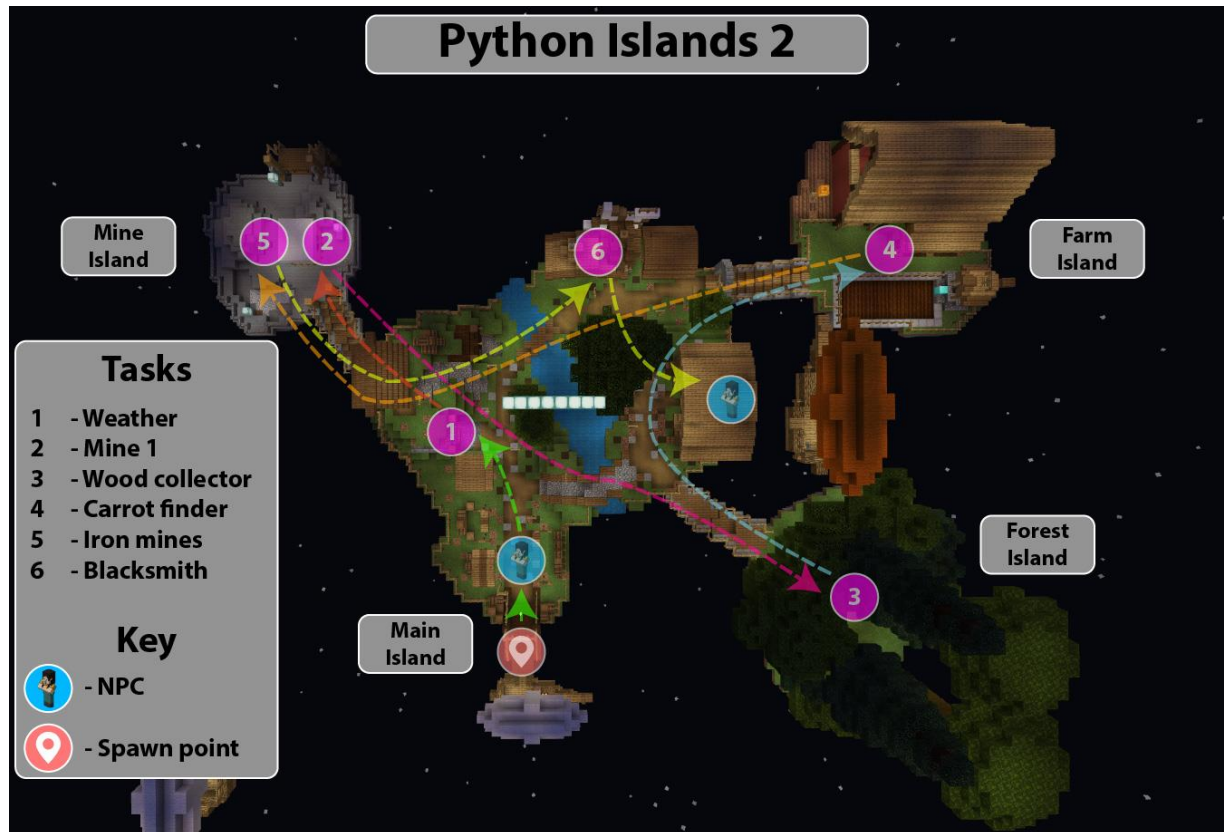
**If Statement** – A decision made in a Python program, based on provided inputs.



# START OF LESSON PROCEDURE

Number of Activities: 5

Optional Activity: 0



## INTRODUCTION AND LEAD-IN: 5 minutes

### Introduction:

Welcome to Python Islands, a place to learn all about the basics of the Python programming language. In Island 2, you will be focusing on decisions in code, specifically “If this, then do this” style decisions.

### Lead-in:

Explain to students, that this island builds upon the previous work they have done in Island 1. Once again, their agent will be returning, although in this lesson they will be focusing on decisions made in code. Their agent will help them complete a series of tasks that they wouldn't be able to otherwise complete.



## CODING ACTIVITIES: 30-45 minutes

### Activity 1: (Weather)

The students start at the landing spot for the island, where they have arrived by small airship. After talking with the town mayor, they are instructed to speak with the scientist. He informs them of a coding concept called “if statements”. The students use this to detect weather changes and output different messages accordingly.

Activity 1 Final Solution
This task does not involve a final solution.

### Activity 2: (Mine 1)

After completing the scientist’s notebook, he opens the door to the mine, allowing access. On reaching the mine, the students will be met by the mine manager.

The manager informs the students that an area of unstable ground has opened up, which he is concerned about. He is especially worried that if he performs tests on the individual areas of ground to check if they are safe, he could fall if the ground subsides. He asks if the student’s agent would be able to test the ground for him, just in case, as the agent can fly.

On inspecting the ground (using a simple if statement), the students find a particular block is unstable and it collapses, opening up what looks to be an old mineshaft.

The mine manager sends the students off to the farm to collect some string and to the forest to collect some wood to make some ladders.

Activity 2 Final Solution
<pre># Final code # ! Must be run twice  agent.move("forward") if check_ground_stable() == False:     say("Unstable ground detected!")     alert()</pre>



### Activity 3: (Forest)

On arrival at the forest, the students are met by the forest explorer. He explains that the forest is covered in fog and students must use their agent to collect wood from up in the tree canopy. The only problem is that they must not disturb the birds' nests. Students use the custom `is_nest()` function to check if there is a nest in front of their agent. If there is not, it is safe to use the custom `harvest()` function to harvest the wood needed. On completion of enough of wood collecting, students are given some of this wood to use for their ladders.

#### Activity 3 Final Solution

```
# Final code
# Run 5 times

broadcast_position()
if (is_nest() == True):
    say("I found a nest!")
    next()
else:
    harvest()
    next()
```

### Activity 4: (Farm)

On arrival at the farm, the students are met by the farm manager. He explains that he has by mistake, mixed some carrot seeds in with his wheat seeds. He has asked if the students can check 4 spots in his field for the type of seed used, marking each with a green or red block, depending on if they are wheat or carrots respectively.

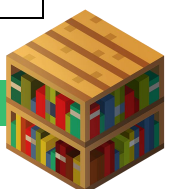
On completion of this task, students are given some string to use for their ladders.

#### Activity 4 Final Solution

```
# Final code
# Run 4 times

agent.move("forward")
agent.move("forward")
agent.move("forward")

block = agent.inspect("down")
say(block)
if(block == "wheat"):
    accept()
else:
```



```
deny()
```

### Activity 5: (Mine 2)

After returning to the mine manager and handing him the materials collected, the mineshaft is opened up and the students can venture down. On arrival in the abandoned mine shaft, students realise there may still be some iron ore left at the end of the shafts. The problem with this though is it is too dark to see far enough down them to know for certain. Given concerns about stability of the mine shaft, it is decided the agent should go down the shafts and check the block at the end. If it is iron ore, it should be extracted. If though it is just stone, it should be left behind, to avoid the mine shaft collapsing.

On collection of three pieces of iron ore, the students are instructed to visit the Blacksmith with this iron ore.

#### Activity 5 Final Solution

```
# Final code
# Run multiple times in front of each mine shaft, then
# use your whistle to move the agent to the front of the
# next mine shaft

agent.move("forward")
agent.move("forward")
agent.move("forward")
agent.move("forward")
agent.move("forward")
agent.move("forward")

if agent.inspect("forward") == "iron_ore":
    agent.destroy("forward")
else:
    say("No iron found!")
```



### Activity 6: (Blacksmith)

On arrival at the blacksmith workshop, students are shown how to check the purity of iron ore. They can then build a program that can automatically use an agent to check each block, selecting only the most pure of iron ore to smelt. This is done using the `purity()` function, which returns back a number, representing the purity of the iron ore. Students then either run `accept()` or `deny()` for that block, to move onto the next block to check.

On completion of this task, the students are provided with some iron ingots to take to the town mayor.

This task focuses on comparing integers in if statements.

Activity 6 Final Solution
<pre># Final code # Run 5 times  if(purity("forward") &lt;= 3):     deny()     say("Purity less than or equal to 3!") else:     say("Purity is acceptable!")     accept()</pre>

### Completion:

Once the student has given the iron ingot to the mayor, the world is complete.

## LESSON CONCLUSION: 5 minutes

Upon completion of this lesson students should be able to answer the following questions:

1. What is the purpose of an if statement?  
Answer: To allow code to branch based on a conditional statement.
2. Complete the following sentence: "In many cases, an if statement is accompanied by an \_\_\_\_\_ statement. This catches anything that doesn't match the conditional statement."  
Answer: Else
3. What does the `==` do within an if statement?  
Answer: Allows comparing if the 2 conditions are equal.
4. How can we check 2 full conditional checks?  
Answer: Using an "and" operator.





## CSTA STANDARDS

The following Computer Science Teachers Association [K-12 Computer Science Standards \(2017\)](#) are covered by this lesson.

Identifier	Standard
1B-AP-09	Create programs that use variables to store and modify data.
1B-AP-11	Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.
1B-AP-12	Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.

