

Educator Guide

Python 101 - Lesson 4

60 minutes

Animals are Friends

LISTS AND METHODS

EDUCATION.MINECRAFT.NET

THEME OVERVIEW

In this lesson, students will help CodingMine with the development of software used to categorize animals for an animal hospital. The software will also assist with the prescribing of treatments and dietary requirements for the animals.

LESSON OBJECTIVES

By the end of the lesson, students will:

- Learn and apply the coding concept of lists
- Understand zero-based numbering
- Use underscores when creating variable and list names
- Spawn mobs in the Minecraft world

KEY VOCABULARY

List – An ordered collection of strings, numerical variables.
Zero-Based Numbering – Counting in computer science that begins with 0
Methods – Special commands that affect data

CODING CONCEPTS

- **List**
  A list is an ordered collection of strings, numerical values or variables. In MakeCode Python, a list can also be a collection of Minecraft items or mobs. A list's contents can be edited by adding, removing, or ordering in a different way. Lists are not unlike variables/boxes; however, in lists, we can place more than one value. When naming lists, the same rules apply as when naming variables. A list name can be anything. However, it must have at least one letter and no spaces. Rather than using spaces, it is common practice to use underscores.
  Example:
  Normal text - Animal list: elephant, lion zebra, giraffe
  Python syntax – `animal_list – [elephant, lion, zebra, giraffe]`

- **Zero-based numbering**
  When we normally count, we start from 1 and continue 2, 3, 4... However, in computer science, and in Python, most counting begins from 0 and continues 1, 2, 3. You will use this concept when coding with lists in this lesson. When you want to access the second value in a list, they will need to refer to it by the numerical value 1.
  `my_list[0]`

- **Methods**
  You will use methods in this lesson to control the contents and the arrangement of content in lists. Methods are special commands that affect data in many ways. Different methods update lists:

  1. `my_list.sort()` – Sorts a list in alphabetical order
  2. `my_list.reverse()` – Reverses the order of the contents
  3. `my_list.append()` – Adds another value to the end of a list
  4. `my_list.pop()` – Removes a value from a list

USER INTERFACE

By the end of the lesson, students will:

- Use the button Q on the keyboard to drop items that they are holding in the hotbar
- See displayed world coordinates in the top left-hand corner

SYNTAX / OPERATORS :

Square Brackets [ ]

Square brackets are used when creating lists. When we want to define a list's contents, they need to be placed within a pair of square brackets and separated by commas. Example:

```
name_list = [John, Mary, Chad, Isabella]
```

Underscore _

When creating names for lists or variables, there are a few guidelines on what is possible and what is not. A list or variable names cannot contain is any spaces. So, if a space is needed the most common thing to use instead is an underscore. Examples:

```
my_list = [0, 1, 2, 3, 4]
important_variable = 1
```

LESSON ACTIVITIES

Direct Instruction (Teacher-Led; "I Do")

Today, we will continue working in our computer science unit, "Python 101". The lesson for today is called "Animals are Friends". (**slide 1**)

Review the lesson objectives with students. (**slide 2**)

Define the important vocabulary with students. (**slide 3**)

“You need to help Coding Mine with their development of new software to be used in the animal hospital. Veterinarians need help to easily identify and categorize animals, as well as their different treatments and dietary requirements. You will use the coding concept of lists to help you in these tasks.” (**slide 4**)

Explain the coding concept focus for students. (**slide 5**)

Explain the new syntax focus for the lesson. (**slide 6**)

Guided Instruction (Teacher Modeling; “We Do”)

Demonstrate how to locate and find the Python 101 lesson, “Animals are Friends” from the in-game library. Showcase the spawn point for students and then explain that they will start their lesson by talking to the NPC, the CEO of CodingMine. (**slides 7-9**)

All students should log into Minecraft: Education Edition at this point and replicate these exact steps so you can complete the first activity together.

Guided Instruction (Teacher Modeling; “We Do”)

Activity 1: Animal Categorizing (Slides 7-16)
Objective: List animals and then spawn them in specific locations

Explanation:
The developer needs your help to make a list of animals and then spawn the different animals’ types in their correct pens, at predefined locations. Start by making a list of the animals going from left to right that matches the list in the Minecraft world.
(Hint: The students should name their list: `my_list`).

When the students run the code correctly, all the animals will spawn in their correct pens and Activity 1 is complete.

Code snippet:

Before:

```
location1 = world(-2, 40, -11)
location2 = world(-2, 40, -5)
location3 = world(-8, 40, -0)
location4 = world(-13, 40, -5)
location5 = world(-13, 40, -11)
# Replace the lines below with your code #

# list of animals

mobs.spawn(My_list[0], location1)
#spawn the third mob from the list at location2
#spawn the fifth mob from the list at location3
#spawn the second mob from the list at location4
#spawn the fourth mob from the list at location5
```

After:

```
location1 = world(-2, 40, -11)
location2 = world(-2, 40, -5)
location3 = world(-8, 40, -0)
location4 = world(-13, 40, -5)
location5 = world(-13, 40, -11)

My_list= [COW, PIG, SHEEP, HORSE, RABBIT]

mobs.spawn(My_list[0], location1)
mobs.spawn(My_list[2], location2)
mobs.spawn(My_list[4], location3)
mobs.spawn(My_list[1], location4)
mobs.spawn(My_list[3], location5)
```

Independent Work (Teacher Support; "You Do")

Activity 2: Dietary Requirement (Slides 17-23)

Objective: Write code for a food making machine for different dogs

Explanation:

The veterinarian needs your help to write some code that will control a machine to make food for three different dogs with diverse dietary requirements.

- The first dog will need everything that is already in the list.
- The second dog will need additional vitamins.

- The third dog will need the beef removed from the list.

You will need to use two methods to manipulate the list of foods: **append** and **pop**. Drop the food into the bowls to feed the dogs.

Each time that the code is run the food will fall into the chest.
(**Hint**: To drop the food into the bowl, tell them to use the key `[Q]` on their keyboard).
(**Hint**: To remove or add any of the items in the list, use the given variables).

Once the students have fed all the dogs, Activity 2 is complete.

Code snippets:

Before:

```
Bone = world(-21, 45, -31)
Beef = world(-21, 45, -29)
Chicken = world(-21, 45, -27)
Biscuit = world(-21, 45, -25)
Vitamins = world(-21, 45, -23)
# Replace the lines below with your code #
Dog_Food=[Bone, Beef, Chicken, Biscuit]
# Add the variable Vitamins to the list using the append method | Step 2
# Remove the variable Beef using the pop method                | Step 3

blocks.place(REDSTONE_BLOCK, Dog_Food[0])
# Change the numerical value of the list below         | Step 1
blocks.place(REDSTONE_BLOCK, Dog_Food[0])
# Change the numerical value of the list below         | Step 1
blocks.place(REDSTONE_BLOCK, Dog_Food[0])
# Change the numerical value of the list below         | Step 1
blocks.place(REDSTONE_BLOCK, Dog_Food[0])
# Change the value of the list below                           | Step 2
blocks.place(REDSTONE_BLOCK, Dog_Food[0])
```

After:

```
Bone = world(-21, 45, -31)
Beef = world(-21, 45, -29)
Chicken = world(-21, 45, -27)
Biscuit = world(-21, 45, -25)
Vitamins = world(-21, 45, -23)
# Replace the lines below with your code #
Dog_Food=[Bone, Beef, Chicken, Biscuit]
# Add the variable Vitamins to the list using the append method | Step 2
# Remove the variable Beef using the pop method                | Step 3

blocks.place(REDSTONE_BLOCK, Dog_Food[0])
blocks.place(REDSTONE_BLOCK, Dog_Food[1])
blocks.place(REDSTONE_BLOCK, Dog_Food[2])
blocks.place(REDSTONE_BLOCK, Dog_Food[3])
# Change the value of the list below                           | Step 2
blocks.place(REDSTONE_BLOCK, Dog_Food[0])
```

```
Bone = world(-21, 45, -31)
Beef = world(-21, 45, -29)
Chicken = world(-21, 45, -27)
Biscuit = world(-21, 45, -25)
Vitamins = world(-21, 45, -23)
# Replace the lines below with your code #
Dog_Food=[Bone, Beef, Chicken, Biscuit]
Dog_Food.append(Vitamins)
#Remove the variable Beef using the pop method               | Step 3

blocks.place(REDSTONE_BLOCK, Dog_Food[0])
blocks.place(REDSTONE_BLOCK, Dog_Food[1])
blocks.place(REDSTONE_BLOCK, Dog_Food[2])
blocks.place(REDSTONE_BLOCK, Dog_Food[3])
blocks.place(REDSTONE_BLOCK, Dog_Food[4])
```

```
Bone = world(-21, 45, -31)
Beef = world(-21, 45, -29)
Chicken = world(-21, 45, -27)
Biscuit = world(-21, 45, -25)
Vitamins = world(-21, 45, -23)
# Replace the lines below with your code #
Dog_Food=[Bone, Beef, Chicken, Biscuit]
Dog_Food.append(Vitamins)
Dog_Food.pop(1)

blocks.place(REDSTONE_BLOCK, Dog_Food[0])
blocks.place(REDSTONE_BLOCK, Dog_Food[1])
blocks.place(REDSTONE_BLOCK, Dog_Food[2])
blocks.place(REDSTONE_BLOCK, Dog_Food[3])
blocks.place(REDSTONE_BLOCK, Dog_Food[4])
```

Activity 3: What is the Cat's Name? (Slides 24-28)
Objective: Write code that uses the sort and reverse methods to a list of cat names

Explanation:
The data scientist needs your help to write a few pieces of code that will use the sort and reverse methods to make changes to a list of cat names that they will be given.

- The first code has to change the last cats name to shadow.
- The second code has to sort the cat names alphabetically.
- The third code has to reverse all of the cat names in the list.

The students will then have to create a code to write out a specific cat name in the chat. This will occur each time, by changing the positional value in the given say command. Then they will have to select that cat from the line-up.

Each time that the code is run, a name of a cat will appear in the chat. To select a cat, tell them to press the button, select a cat underneath the monitor, and then to choose that cat from the line-up by pressing the button underneath the correct cat. When the students have selected all of the cats, Activity 3 and the lesson are complete.

Code Snippets:

Before:

```
Cat_Names= ["Smokey", "Oreo", "Sammy", "Patch", "Princess", "Snowy"]
# Replace the lines below with your code #
#Change the last name to shadow           | Step 1
#Sort the names alphabetically in the list        | Step 2
#Reverse all names                                      | Step 3

# Change the value of the list below | Step 1,2,3
player.say(Cat_Names[])
```

After:

```
Cat_Names= ["Smokey", "Oreo", "Sammy", "Patch", "Princess", "Shadow"]
# Replace the lines below with your code #
Cat_Names[5] = "Shadow"
#Sort the names alphabetically in the list        | Step 2
#Reverse all names                                      | Step 3

# Change the value of the list below | Step 2,3
player.say(Cat_Names[5])
```

```
Cat_Names= ["Smokey", "Oreo", "Sammy", "Patch", "Princess", "Shadow"]
# Replace the lines below with your code #
Cat_Names[5] = "Shadow"
Cat_Names.sort()
#Reverse all names                                          | Step 3

# Change the value of the list below | Step 3
player.say(Cat_Names[3])
```

```
Cat_Names= ["Smokey", "Oreo", "Sammy", "Patch", "Princess", "Shadow"]

Cat_Names[5] = "Shadow"
Cat_Names.sort()
Cat_Names.reverse()

player.say(Cat_Names[3])
```

LESSON CONCLUSION

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

1. What is a list?
   Answer: A list is a collection of strings, numerical values, variables, mobs or items.
2. What do we use instead of spaces when naming a list or variable?
   Answer: Underscore
3. What syntax do you place around a list's content?
   Answer: Square brackets
4. What four methods can we use with lists?
   Answer: Reverse, append, sort, pop

These questions are also available as a printable handout at the end of this document. They can be used as a formative assessment for this lesson's learning objectives.

EDUCATIONAL STANDARDS

UNITED STATES: CSTA

- **3A-AP-14** Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.
- **3A-AP-18** Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

UNITED STATES: ISTE

- **1.5.a** Students formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions.
- **1.5.c** Students break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.

AUSTRALIAN F-10 CURRICULUM: DIGITAL TECHNOLOGIES (YEAR 9 AND 10)

- Designing algorithms to solve real-world problems and describing algorithms using flow charts and structured English (ACTDIP040)
- Recognising that different algorithms can solve a problem with different trade-offs (ACTDIP040)
- Considering different algorithms and selecting the most appropriate based on the type of problem (ACTDIP041)

UK NATIONAL CURRICULUM: COMPUTING (KEY STAGE 4)

- Develop and apply their analytic, problem-solving, design, and computational thinking skills

NAME: _____ **DATE:** _____

PYTHON 101: LESSON 4 FORMATIVE ASSESSMENT

What is a list?	
What do we use instead of spaces when naming a list or variable?	
What syntax do you place around a list's content?	
What four methods can we use with lists?	