

# AI & Robotics Lab – Week 6

## Topic: Lists, Dictionaries, Loops & Data Comparison in Python

**Duration:** 2 Hour

**Instructor:** Azeem Aslam

**Institute:** UVAS Pattoki

## Learning Outcomes

- Understand what **Lists** and **Dictionaries** are in Python
- Create and modify lists and dictionaries
- Use **loops** to repeat tasks automatically
- Compare data using **if, elif, else** conditions
- Apply these concepts in small AI or Robotics examples

## Concept 1: Lists in Python

Think of a **list** as a *box with compartments* where you can keep many related items together (students, robots, sensors).

### Key Points

- Lists are **ordered** (items stay in sequence)
- Lists are **changeable** (we can add or remove items)
- Lists can hold **different types** (numbers, strings, etc.)

Syntax example: `robots = ["Alpha", "Beta", "Gamma"]`

```
In [1]: # Example 1 – Creating and Accessing Lists
# A list is defined with square brackets [] and items separated by commas.
```

```
robots = ["Alpha", "Beta", "Gamma"]

print("Robot List:", robots)          # print the whole list
print("First Robot:", robots[0])      # index 0 = first item
print("Second Robot:", robots[1])
print("Last Robot:", robots[-1])     # -1 = last item from the end
```

Robot List: ['Alpha', 'Beta', 'Gamma']  
First Robot: Alpha  
Second Robot: Beta  
Last Robot: Gamma

```
In [2]: # Example 2 - Modifying a List
# .append() adds to the end, .remove(value) deletes the first matching item.
robots.append("Delta")                # add a new robot
robots.remove("Beta")                 # remove a robot named 'Beta'
print("Updated Robot List:", robots)
```

Updated Robot List: ['Alpha', 'Gamma', 'Delta']

## ✓ Try It Yourself

1. Create a list named `fruits` with 5 fruit names.
2. Print the **first** and **last** fruit.
3. Add a new fruit and remove one old fruit.

Write your code in the cell below:

```
In [ ]: # TODO: Your practice for Lists goes here
fruits = [] # e.g., ["apple", "banana", "mango", "orange", "grapes"]

# 1) print first and last fruit
# 2) add and remove an item
# 3) print the final list
```

## Concept 2: Dictionaries in Python

A **dictionary** stores data as **key-value pairs**. Think of it like a **robot record card**.

Example: `{ "name": "Alpha", "battery": 80, "status": "Active" }`

**Why use a dictionary?** Because we can access information by its **key** quickly.

```
In [3]: # Example 3 - Creating and Accessing a Dictionary
robot = {"name": "Alpha", "battery": 80, "status": "Active"}

print("Robot Name:", robot["name"])    # access by key
print("Battery Level:", robot["battery"])
print("Status:", robot["status"])
```

Robot Name: Alpha

Battery Level: 80

Status: Active

```
In [4]: # Example 4 - Updating a Dictionary
robot["battery"] = 95                # update a value
robot["task"] = "Line Following"     # add a new key-value pair
print("Updated Robot Info:", robot)
```

Updated Robot Info: {'name': 'Alpha', 'battery': 95, 'status': 'Active', 'task': 'Line Following'}

## ✓ Try It Yourself

Create a dictionary for a student with keys: `name`, `roll_no`, and `score`.

Increase `score` by 5 and add a new key `grade`.

Write your code in the cell below:

```
In [ ]: # TODO: Your practice for dictionaries goes here
student = {}
# Example structure:
# student = {"name": "Ali", "roll_no": 7, "score": 72}

# 1) increase score by 5
# 2) add a new key 'grade'
# 3) print the final dictionary
```



## Concept 3: Loops in Python

Loops repeat actions automatically. Use **for** when you know how many times, and **while** when you repeat *until* a condition changes.

```
In [5]: # For Loop - iterate over a list
        robots = ["Alpha", "Gamma", "Delta"]
        for r in robots:
            print("Activating Robot:", r)
```

Activating Robot: Alpha

Activating Robot: Gamma

Activating Robot: Delta

```
In [6]: # While Loop - run until battery becomes 0
        battery = 5
        while battery > 0:
            print("Battery remaining:", battery)
            battery -= 1 # decrease by 1
```

Battery remaining: 5

Battery remaining: 4

Battery remaining: 3

Battery remaining: 2

Battery remaining: 1



### Try It Yourself

1. Using a **for loop**, print each character of the string "ROBOT".
2. Using a **while loop**, count down from 3 to 1 and then print "Go!".

```
In [ ]: # TODO: Your practice for Loops goes here
        # 1) for loop over a string
        # 2) while loop countdown
```



## Concept 4: Data Comparison (Conditions)

Use `if`, `elif`, and `else` to let your code **make decisions** like a simple AI.

```
In [7]: # Example - Battery-based Decision
battery = 40

if battery > 60:
    print("Battery is Full - Ready for Mission!")
elif battery > 30:
    print("Battery is Medium - Prepare for Recharge.")
else:
    print("Battery Low - Returning to Base.")
```

Battery is Medium - Prepare for Recharge.

## ✓ Try It Yourself

Take a variable `temp = 38`.

If `temp > 40` print **"Alert: High Temperature!"**, else print **"Normal"**.




```
In [ ]: # TODO: Your practice for conditions goes here
temp = 38
# Write your if-else decision here
```

## ✂ Mini AI Project: Robot Battery Checker

You have 3 robots. Each robot has a battery level.

Use a **for loop** + **if condition** to print whether each robot can continue working or needs charging.

```
In [8]: # Mini Project - Robot Battery Checker
# .items() Lets us loop over (key, value) pairs in a dictionary
robots = {"Alpha": 75, "Beta": 20, "Gamma": 60}
for name, battery in robots.items():
    if battery > 50:
        print(name, "→ Battery OK ✓ (", battery, "%)")
    else:
        print(name, "→ Low Battery ⚠ (", battery, "%)")
```

Alpha → Battery OK  ( 75 %)  
Beta → Low Battery  ( 20 %)  
Gamma → Battery OK  ( 60 %)

## Quick Revision Table

Concept	What It Does	Example
<b>List</b>	Stores many items in order	<code>["Alpha", "Beta"]</code>
<b>Dictionary</b>	Stores key–value pairs	<code>{ "name": "Alpha" }</code>
<b>For Loop</b>	Repeats fixed number of times	<code>for x in list:</code>
<b>While Loop</b>	Repeats until condition false	<code>while x &gt; 0:</code>
<b>If-Else</b>	Makes decisions	<code>if battery &gt; 60:</code>

## Reflection Questions

1. What is the difference between a **list** and a **dictionary**?
2. When should you use a **for** loop vs a **while** loop?
3. Why do we need **if/else** conditions in robotics or AI?

## Homework Challenge

Write a program that:

1. Takes 5 temperature readings in a list.
2. Prints each temperature using a **for** loop.
3. If any reading is above **40°C**, print **"Alert: High Temperature!"**