

21-Day Python Challenge – Day 7

Lists in Python

Day 7 Learning Objectives

By the end of today's class, students will be able to:

- Understand what a **List** is.
 - Create a List.
 - Access List elements.
 - Update List elements.
 - Delete List elements.
 - Traverse (Loop through) a List.
 - Use commonly used List Methods.
 - Build simple real-world List programs.
-

What is a List?

A **List** is a collection of multiple values stored in a single variable.

Instead of creating many variables,

```
student1 = "Rahul"  
student2 = "Ravi"  
student3 = "Anitha"  
student4 = "Priya"
```

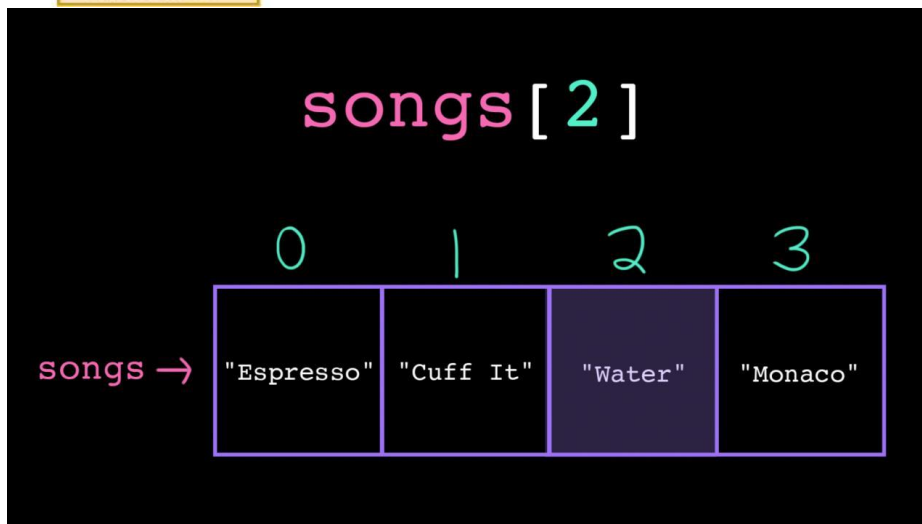
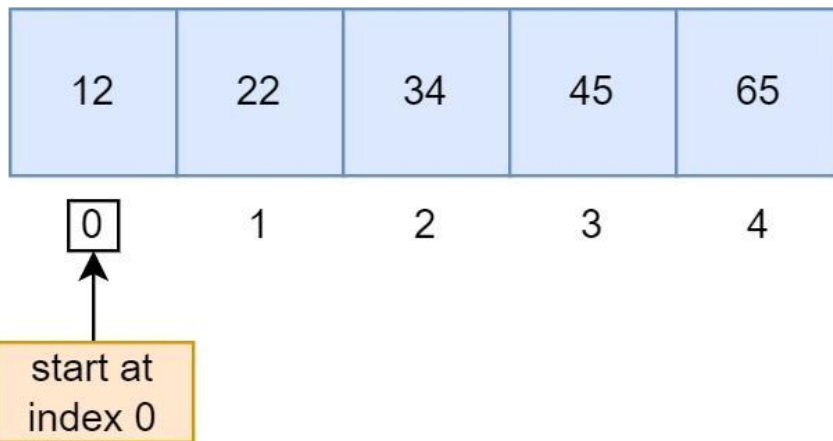
We can write

```
students = ["Rahul", "Ravi", "Anitha", "Priya"]
```

This is called a **List**.

Picture 1 – What is a List?

List



Think of a list like a **school attendance register**.

Students

```
+-----+
| Rahul  |
| Ravi   |
| Anitha |
| Priya  |
+-----+
```

Why Do We Use Lists?

Lists help us store:

- Student Names
- Employee Details
- Marks
- Mobile Numbers
- Products
- Books
- Cities

Instead of creating hundreds of variables.

Creating a List

```
students = ["Rahul", "Ravi", "Anitha", "Priya"]  
print(students)
```

Output

```
['Rahul', 'Ravi', 'Anitha', 'Priya']
```

List Indexing

Every item has an **Index Number**.

```
students = ["Rahul", "Ravi", "Anitha", "Priya"]
```

Student Rahul Ravi Anitha Priya

Index 0 1 2 3



List name

List elements

SCALER
Topics

```
#Accessing elements of list
my_list = [1, 2, 3, 4, 5]
```

```
# Accessing the first element (index 0)
print(my_list[0])
```

Output: 1

```
# Accessing the third element (index 2)
print(my_list[2])
```

Output: 3

```
# Accessing the last element (index -1)
print(my_list[-1])
```

Output: 5



Index

0

1

2

3

countryList

"India"

"China"

"Nepal"

"Bhutan"

```
countryList[3]
```

```
countryList[4]
```

Bhutan

IndexError: list index out of range

We are trying to access index
which does not exist

Accessing List Items

```
students = ["Rahul", "Ravi", "Anitha", "Priya"]  
  
print(students[0])  
print(students[2])
```

Output

```
Rahul  
Anitha
```

Part 1 – Create a List

```
fruits = ["Apple", "Mango", "Orange"]  
  
print(fruits)
```

Practical 1

```
colors = ["Red", "Green", "Blue"]  
  
print(colors)
```

Part 2 – Update List

Suppose Rahul changes his name to Rakesh.

```
students = ["Rahul", "Ravi", "Anitha"]  
  
students[0] = "Rakesh"  
  
print(students)
```

Output

```
['Rakesh', 'Ravi', 'Anitha']
```

Practical 2

```
cities = ["Vizag", "Hyderabad", "Chennai"]  
  
cities[1] = "Bangalore"  
  
print(cities)
```

Part 3 – Delete List Items

Use **del**

```
students = ["Rahul", "Ravi", "Anitha"]  
  
del students[1]  
  
print(students)
```

Output

```
['Rahul', 'Anitha']
```

Part 4 – Traverse a List

Traverse means

Visit every item one by one.

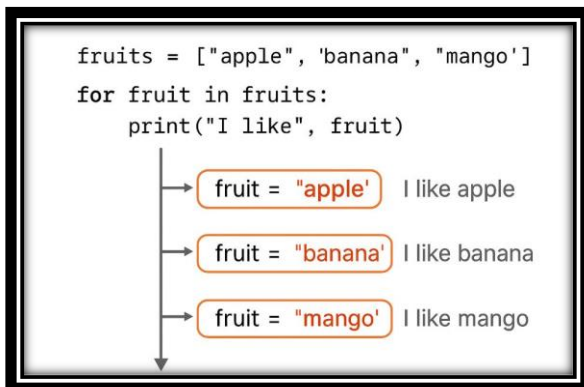
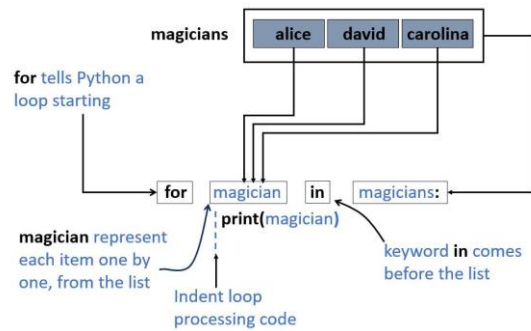
Use **for loop**

```
students = ["Rahul", "Ravi", "Anitha", "Priya"]  
  
for student in students:  
    print(student)
```

Output

```
Rahul  
Ravi  
Anitha  
Priya
```

Picture 3 – Traversing a List



Start the for Loop

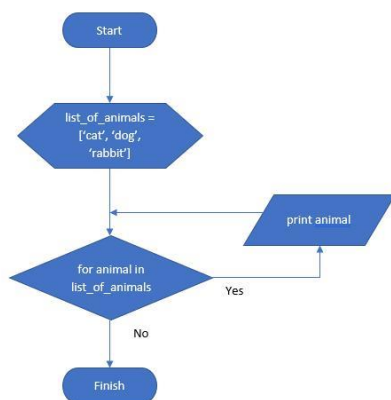
```
for fruit in fruits:
```

Break it into parts.

Word	Meaning
for	Repeat the code
fruit	Temporary variable
in	Take one item at a time
fruits	The list

Python understands this as:

"Take one fruit from the list, store it in the variable `fruit`, and repeat until the list ends."



Think of a teacher calling attendance.

Rahul

↓

Ravi

↓

Anitha

↓

Priya

Part 5 – List Methods

Python provides many useful methods.

append()

Adds an item at the end.

```
students = ["Rahul", "Ravi"]  
students.append("Anitha")  
print(students)
```

Output

```
['Rahul', 'Ravi', 'Anitha']
```

insert()

Adds an item at a specific position.

```
students = ["Rahul", "Anitha"]  
students.insert(1, "Ravi")  
print(students)
```

Output

```
['Rahul', 'Ravi', 'Anitha']
```

remove()

Removes an item by name.

```
students = ["Rahul", "Ravi", "Anitha"]  
students.remove("Ravi")  
print(students)
```

Output

```
['Rahul', 'Anitha']
```

pop()

Removes the last item.

```
students = ["Rahul", "Ravi", "Anitha"]  
students.pop()  
print(students)
```

Output

```
['Rahul', 'Ravi']
```

sort()

Sorts alphabetically.

```
students = ["Ravi", "Rahul", "Anitha"]  
students.sort()  
print(students)
```

Output

```
['Anitha', 'Rahul', 'Ravi']
```

reverse()

Reverses the list.

```
numbers = [1,2,3,4,5]  
numbers.reverse()  
print(numbers)
```

Output

```
[5,4,3,2,1]
```

len()

Counts total items.

```
students = ["Rahul", "Ravi", "Anitha"]  
print(len(students))
```

Output : 3

Real-Time Program 1

Student Attendance

```
students = ["Rahul","Ravi","Anitha","Priya"]  
  
print("Today's Attendance")  
  
for student in students:  
    print(student)
```

Real-Time Program 2

Shopping List

```
shopping = ["Milk","Rice","Sugar"]  
  
shopping.append("Oil")  
  
for item in shopping:  
    print(item)
```

Real-Time Program 3

Top Students

```
students = ["Rahul","Ravi","Anitha"]  
  
students.sort()  
  
for student in students:  
    print(student)
```

Real-Time Program 4

Mobile Contacts

```
contacts = ["Father","Mother","Friend"]  
  
contacts.append("Teacher")  
  
for contact in contacts:  
    print(contact)
```

Real-Time Program 5

Favourite Subjects

```
subjects = ["Python", "Maths", "Science"]  
subjects.remove("Maths")  
for subject in subjects:  
    print(subject)
```

Mini Project

Student Management System

```
students = []  
count = int(input("Enter Number of Students : "))  
for i in range(count):  
    name = input("Enter Student Name : ")  
    students.append(name)  
print("\nStudent List")  
for student in students:  
    print(student)  
print("\nTotal Students :", len(students))
```

Practice Programs

1. Create a list of five fruits.
2. Print all fruits.
3. Change one fruit name.
4. Delete one fruit.
5. Add a new fruit.
6. Print all fruits using a loop.
7. Sort the list.
8. Reverse the list.
9. Count total fruits.
10. Create a shopping list.

Assignment

Write Python programs to:

- Create a student list.
- Update a student's name.
- Delete a student.
- Traverse the list.
- Add new students using `append()`.
- Insert a student at a specific position.
- Remove a student by name.
- Sort students alphabetically.
- Reverse the student list.
- Count the total students.

Interview Questions

1. What is a List?
2. Why do we use Lists?
3. Which brackets are used to create a List?
4. How do you access a List item?
5. How do you update a List item?
6. How do you delete a List item?
7. What does `append()` do?
8. What is the difference between `append()` and `insert()`?
9. What does `remove()` do?
10. How do you traverse a List?

Day 7 Challenge

Student Marks List

Write a program that:

- Creates an empty list.
- Accepts marks for **5 students**.
- Stores the marks in the list.
- Displays all the marks.
- Finds and displays:
 - Total number of marks entered
 - Highest mark
 - Lowest mark
 - Average mark

Hint: You can use `append()`, `len()`, `max()`, `min()`, and `sum()`.

Day 7 Success Message

Congratulations! 🎉

Today you learned one of the most useful data structures in Python—**Lists**. Lists allow you to store, organize, update, and process multiple pieces of information efficiently.

You explored:

- ☒ Creating Lists
- ☒ Accessing elements using indexes
- ☒ Updating and deleting items
- ☒ Traversing with a `for` loop
- ☒ Using common List methods like `append()`, `insert()`, `remove()`, `pop()`, `sort()`, `reverse()`, and `len()`

"A variable stores one value. A list stores many values. Learning lists is your first step toward building real-world Python applications such as student management systems, shopping carts, contact books, and inventory software."