**Module 8-** Lists

Contents

[**8**](#_heading=h.ls088s34ovkm) Lists…………………………………………………………………………………………………………………………………………………. 1

[8.1](#_heading=h.2vrvk7f0fhjt) Creating a List………………………………………………………………………………………………………………………… 1

[8.2](#_heading=h.s0jht0e28pbp) Accessing List Elements…………………………………………………………………………………………………………..1

[8.3](#_heading=h.7b7h59cu86g0) Modifying Lists………………………………………………………………………………………………………………………. 2

[8.4](#_heading=h.oe9hmaevvryo) Removing Elements……………………………………………………………………………………………………………….. 2

[8.5](#_heading=h.3mr4nln1dh8k) Iterating Through Lists……………………………………………………………………………………………………………. 2

[8.6](#_heading=h.4lecxbuuz42h) List Slicing………………………………………………………………………………………………………………………………. 2

[8.7](#_heading=h.r04do5k5sovz) Common List Methods…………………………………………………………………………………………………………… 3

[8.8](#_heading=h.fhry506m65rl) Nested Lists……………………………………………………………………………………………………………………………. 3

# Lists

Lists are essential data structures in Python, enabling storage of ordered and mutable item collections. This chapter offers a thorough overview of Python lists, covering creation, access, modification, and manipulation techniques.

* 1. Creating a List

To [create a list](https://www.geeksforgeeks.org/python/python-lists/), enclose elements in square brackets [], separated by commas. Python lists can hold any data type, such as integers, strings, floats, or even other lists.

|  |
| --- |
| fruits = [“apple”, “banana”, “cherry”]  numbers = [1, 2, 3, 4, 5]  mixed = [42, “hello”, True, 3.14] |

To [create an empty list](https://www.freecodecamp.org/news/python-empty-list-tutorial-how-to-create-an-empty-list-in-python/):

|  |
| --- |
| empty\_list = [] |

* 1. Accessing List Elements

You can [access list elements](https://www.w3schools.com/python/python_lists_access.asp) using their index, starting at 0. Python also allows negative indexing, with -1 indicating the last element.

|  |
| --- |
| print(fruits[0])  print(fruits[-1]) |

* 1. Modifying Lists

Python [lists are mutable](https://www.geeksforgeeks.org/python/concept-of-mutable-lists-in-python/), so you can [modify their contents after creation](https://builtin.com/data-science/python-list). To change a value, assign a new one to a specific index.

|  |
| --- |
| fruits[1] = “blueberry”  print(fruits) |

Add elements using [append()](https://www.w3schools.com/python/ref_list_append.asp), [insert()](https://www.programiz.com/python-programming/methods/list/insert), or [extend()](https://www.geeksforgeeks.org/python/python-list-extend-method/):

|  |
| --- |
| fruits.append(“orange”)  fruits.insert(1, “kiwi”)  fruits.extend([“grape”, “melon”]) |

* 1. Removing Elements

You can [delete items from a list](https://www.w3schools.com/python/python_lists_remove.asp) using [remove(), pop(), or the del statement](https://note.nkmk.me/en/python-list-clear-pop-remove-del/).

|  |
| --- |
| fruits.remove(“apple”)  fruits.pop()  del fruits[0] |

remove() deletes the first occurrence of a value, while pop() removes and returns an element, with the default being the last element.

* 1. Iterating Through Lists

You can [iterate over lists](https://www.geeksforgeeks.org/python/iterate-over-a-list-in-python/) using for loops or while loops. The enumerate() function allows you to access both the index and the value.

|  |
| --- |
| for fruit in fruits:  print(fruit)  for index, value in enumerate(fruits):  print(index, value) |

* 1. List Slicing

[Slicing](https://www.pythonmorsels.com/slicing/) allows you to extract a part of a list with the syntax list[start:stop:step].

|  |
| --- |
| numbers = [0, 1, 2, 3, 4, 5]  print(numbers[1:4])  print(numbers[:3])  print(numbers[::2]) |

* 1. Common List Methods

Python lists come with built-in methods that facilitate data manipulation. [Some frequently used methods are](https://docs.python.org/3/tutorial/datastructures.html):

* append(x): Add an item to the end

|  |
| --- |
| fruits = ["apple", "banana"]  fruits.append("cherry")  print(fruits) # Output: ['apple', 'banana', 'cherry'] |

* extend(iterable): Add multiple items

|  |
| --- |
| fruits = ["apple", "banana"]  fruits.extend(["cherry", "orange"])  print(fruits) # Output: ['apple', 'banana', 'cherry', 'orange'] |

* insert(i, x): Insert at a given index

|  |
| --- |
| fruits = ["apple", "banana"]  fruits.insert(1, "kiwi")  print(fruits) # Output: ['apple', 'kiwi', 'banana'] |

* remove(x): Remove first occurrence

|  |
| --- |
| fruits = ["apple", "banana", "apple"]  fruits.remove("apple")  print(fruits) # Output: ['banana', 'apple'] |

* pop([i]): Remove and return item

|  |
| --- |
| fruits = ["apple", "banana", "cherry"]  popped = fruits.pop(1)  print(popped) # Output: banana  print(fruits) # Output: ['apple', 'cherry'] |

* clear(): Remove all items

|  |
| --- |
| fruits = ["apple", "banana"]  fruits.clear()  print(fruits) # Output: [] |

* index(x): Return index of item

|  |
| --- |
| fruits = ["apple", "banana", "cherry", "banana"]  print(fruits.index("banana")) # Output: 1  print(fruits.index("banana", 2)) # Output: 3 |

* count(x): Count occurrences

|  |
| --- |
| fruits = ["apple", "banana", "apple"]  print(fruits.count("apple")) # Output: 2 |

* sort(): Sort in place

|  |
| --- |
| numbers = [3, 1, 4, 2]  numbers.sort()  print(numbers) # Output: [1, 2, 3, 4]  numbers.sort(reverse=True)  print(numbers) # Output: [4, 3, 2, 1] |

* reverse(): Reverse the list

|  |
| --- |
| numbers = [1, 2, 3]  numbers.reverse()  print(numbers) # Output: [3, 2, 1] |

* len(): return the number of items in the list

|  |
| --- |
| fruits = ["apple", "banana"]  print(len(fruits)) # Output: 2 |

* 1. Nested Lists

[Lists can include other lists](https://www.freecodecamp.org/news/list-within-a-list-in-python-initialize-a-nested-list/). Access items with multiple indices:

|  |
| --- |
| matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  print(matrix[0][1]) |

Programming Exercises:

1. **Creating Lists:**

Create a list called colors containing the following strings: "red", "green", "blue". Then print the list.

1. **Accessing Elements:**

Create a list of at least 5 colors, then print the first and last elements using indexing.

1. **Modifying Lists**

Using the colors list from exercise 1, change “green” to “purple” and append “orange” to the end, then print.

1. **Removing Elements**

Given the list

fruits = [“apple”, “banana”, “cherry”, “banana”]

Remove the first occurrence of “banana” and remove the last element using pop(), then print the list.

1. **Iterating Through Lists**

Create a list with 3 elements. Write a **for** loop that prints each element followed by the index of that element using enumerate()

1. **List Slicing**

Create a list with the elements being integers 0-9. Print a slice containing numbers from index 3 to 8. Then print every other integer using slicing.

1. **Searching Lists**

Using the list

fruits = [“apple”, “banana”, “apple”, “cherry”]

Count how many times “apple” appears and find the index of “cherry”.

1. **Sorting and Reversing**

Given the list

numbers = [5, 2, 9, 1, 5, 6]

Sort the list in ascending order and print, the reverse the sorted list and print that.

1. **Nested Lists**

Create a nested list called matrix that represents a 3x3 grid, use integers 1-9 like on a number pad. Print the element in the second row, third column. Change the element in the first row, first column to 0. Print the new matrix.

1. **List Manipulation**

Use the list:

data = [3, 7, 2, 3, 7, 8, 2]

Remove all duplicate values and sort the list in descending order, then print.

Programming Exercise Solutions

1.

|  |
| --- |
| colors = ["red", "green", "blue"]  print(colors) |

2.

|  |
| --- |
| colors = ["red", "green", "blue", “orange”, “purple”]  print(colors[0])  print(colors[-1] |

3.

|  |
| --- |
| colors = ["red", "green", "blue"]  colors[1] = "purple"  colors.append("orange")  print(colors) |

4.

|  |
| --- |
| fruits = ["apple", "banana", "cherry", "banana"]  fruits.remove("banana")  fruits.pop()  print(fruits) |

5.

|  |
| --- |
| elements = [“e1”, “e2”, “e3”]  for index, element in enumerate(elements):  print(element, index) |

6.

|  |
| --- |
| numbers = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  print(numbers[3:9])  print(numbers[::2]) |

7.

|  |
| --- |
| words = ["apple", "banana", "apple", "cherry"]  print(words.count("apple"))  print(words.index("cherry")) |

8.

|  |
| --- |
| numbers = [5, 2, 9, 1, 5, 6]  numbers.sort()  print(numbers)  numbers.reverse()  print(numbers) |

9.

|  |
| --- |
| matrix = [  [1, 2, 3],  [4, 5, 6],  [7, 8, 9]  ]  print(matrix[1][2])  matrix[0][0] = 0  print(matrix) |

10.

|  |
| --- |
| data = [3, 7, 2, 3, 7, 8, 2]  unique\_data = []  for item in data:  if item not in unique\_data:  unique\_data.append(item)  unique\_data.sort(reverse=True)  print(unique\_data) |