

# Queues

## Introduction

A **queue** is a fundamental concept in computer science, used for storing and managing data in a specific order. Here are the key points:

- **Definition:** A queue is a linear data structure where elements are stored following the **First In, First Out (FIFO)** principle. This means that the first element inserted is the first one to be removed.
- **Operations:**
  - **Enqueue:** Adds an element to the back of the queue.
  - **Dequeue:** Removes and returns the front element from the queue.
  - **Front:** Returns the front element without removing it.
  - **Rear:** Returns the last (most recently added) element.
  - **isEmpty:** Checks if the queue is empty.
  - **isFull:** Checks if the queue is full (though this is less common).
- **Representation:** Imagine a line of people waiting to purchase tickets—the person at the front is served first (FIFO). The front of the queue is where removal occurs, and the rear is where new elements are added.
- **Types of Queues:**
  - **Simple Queue:** Follows a straightforward FIFO structure.
  - **Double-Ended Queue (Deque):** Allows insertion and deletion from both ends.
  - **Circular Queue:** Connects the last position back to the first position.
- **Applications:** Queues are used in scenarios like task scheduling, print spooling, and breadth-first search algorithms.

Remember, a queue is like that patient line at the doctor's office—first come, first served! <sup>123</sup>

## Implementation

### Python Usage

```
from collections import deque

# Create an empty deque
queue = deque()

# Enqueue (add) items to the back of the queue
queue.append('apple')
queue.append('banana')
queue.append('cherry')

# Dequeue (remove) an item from the front of the queue
front_item = queue.popleft()
```

```

print("Dequeued item:", front_item)  # Should print 'apple'

# Check if the queue is empty
print("Is queue empty?", len(queue) == 0)  # Should print False

# Access the front item without removing it
print("Front item:", queue[0])  # Should print 'banana'

```

## C++ Usage

```

#include <iostream>
#include <queue>

int main() {
    // Create a queue of integers
    std::queue<int> myQueue;

    // Enqueue (add) elements to the back of the queue
    myQueue.push(10);
    myQueue.push(20);
    myQueue.push(30);

    // Print the front element (without removing it)
    std::cout << "Front element: " << myQueue.front() << '\n';

    // Dequeue (remove) the front element
    myQueue.pop();

    // Print the new front element
    std::cout << "New front element: " << myQueue.front() << '\n';

    // Check if the queue is empty
    std::cout << "Is queue empty? " << (myQueue.empty() ? "Yes" : "No") << '\n';

    // Get the size of the queue
    std::cout << "Queue size: " << myQueue.size() << '\n';

    return 0;
}

```