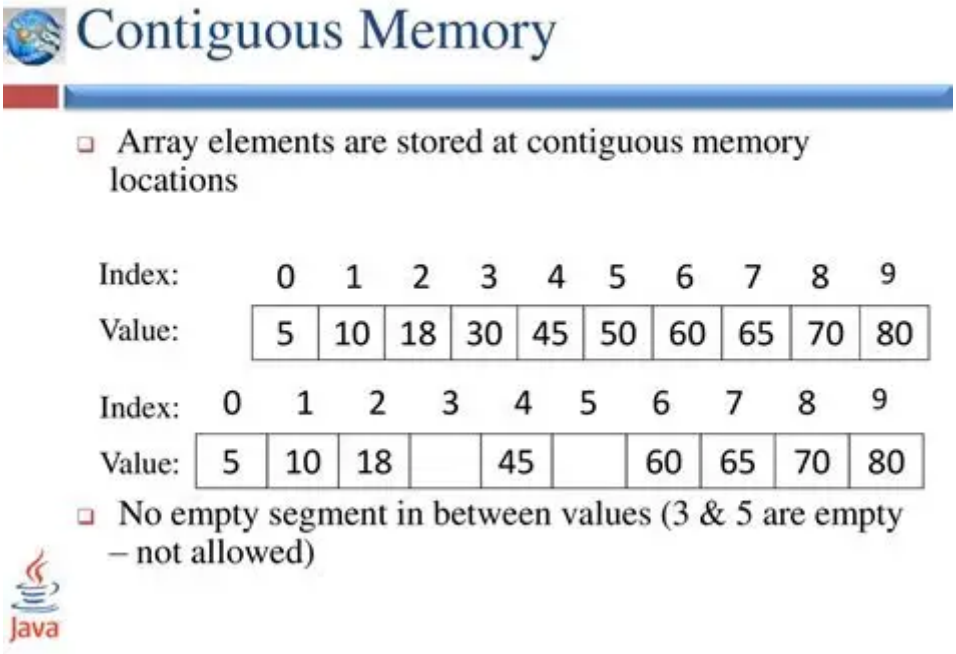
1. Introduction: From Python Lists to C Arrays

1.1 Key Differences Overview

Aspect	Python Lists	C Arrays
Size	Dynamic (can grow/shrink)	Fixed size (static)
Type	Can store mixed types	Single type only
Memory	Automatic management	Manual bounds checking
Declaration	`list = [1, 2, 3]`	`int arr[5] = {1, 2, 3, 4, 5};`
Bounds Checking	Automatic (raises IndexError)	No automatic checking
Performance	Slower (overhead)	Faster (direct memory access)

1.2 Why Arrays Matter in C



Memory Efficiency:

- Arrays store elements in contiguous memory locations

- Faster access compared to dynamic structures
- Predictable memory usage

Performance:

- Direct indexing without function call overhead
- Cache-friendly memory access patterns
- Essential for embedded systems and real-time applications

Revision #2
Created 2025-09-15 00:50:24 UTC by DS
Updated 2025-09-15 00:51:54 UTC by DS