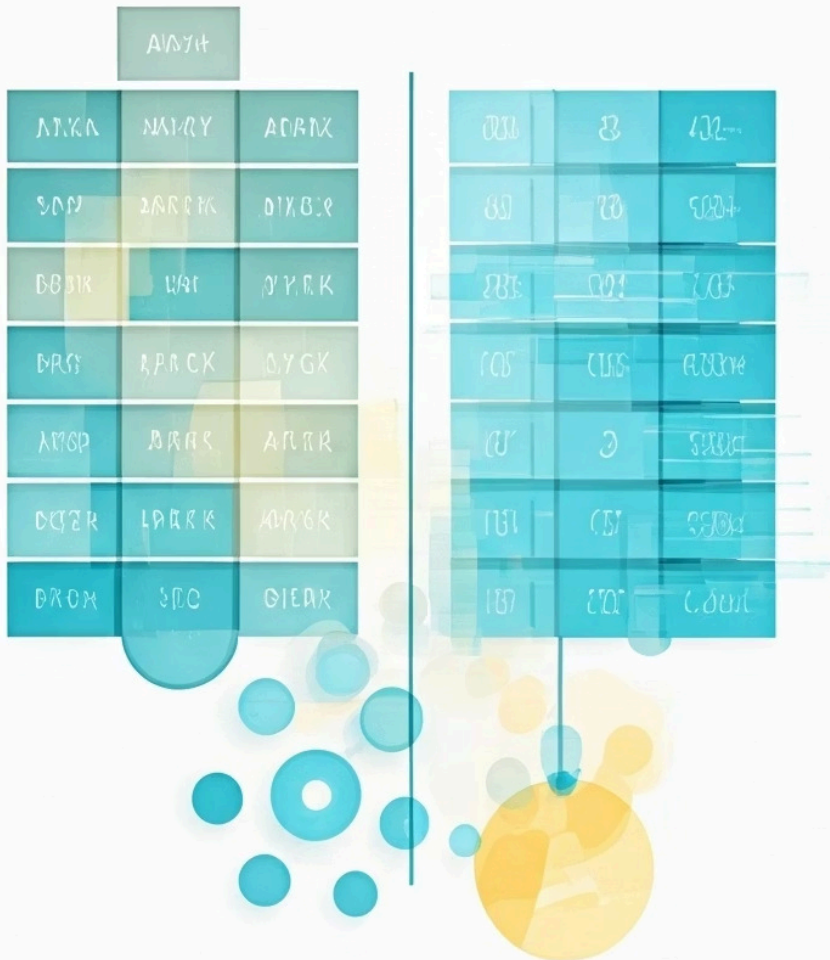




Class 9

Arrays vs Lists, Tuples & Practice Questions

Arrays vs Lists



Arrays

Same data type, contiguous memory

Memory-efficient

- Example: A NumPy array of 5 integers might look like this in memory:

css

[Copy code](#)

```
[ 4 bytes ][ 4 bytes ][ 4 bytes ][ 4 bytes ][ 4 bytes ]  
int0      int1      int2      int3      int4
```

Lists

Different types, dynamic sizing

More flexible

```
[ ptr0 ][ ptr1 ][ ptr2 ][ ptr3 ]  
  ↓     ↓     ↓     ↓  
objA  objB  objC  objD  (objects are elsewhere)
```

Arrays: Common Methods



Create & Add

`import numpy as np`

`arr = np.array('i', [1, 2, 3])`

`arr.append(4)`



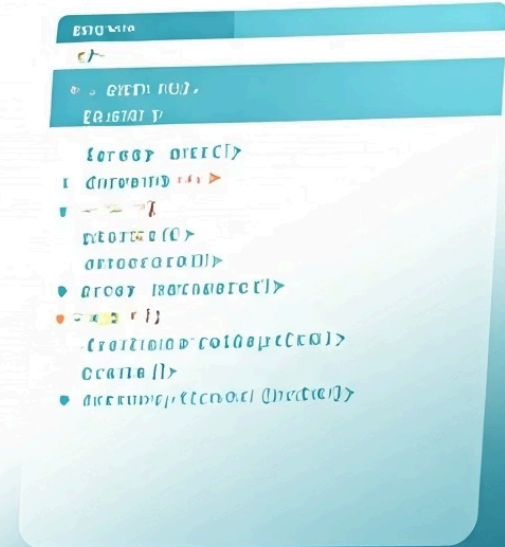
Access & Remove

`arr[0], arr.pop(2)`



Count

`arr.count(3)`



Lists: Versatile & Dynamic



Mixed types: `[1, "apple", 3.14]`

Methods: `.append()`, `.insert()`, `.remove()`

Negative indexing: `list[-1]` → last element

Stores pointers to objects

Why Choose Arrays or Lists?

Arrays

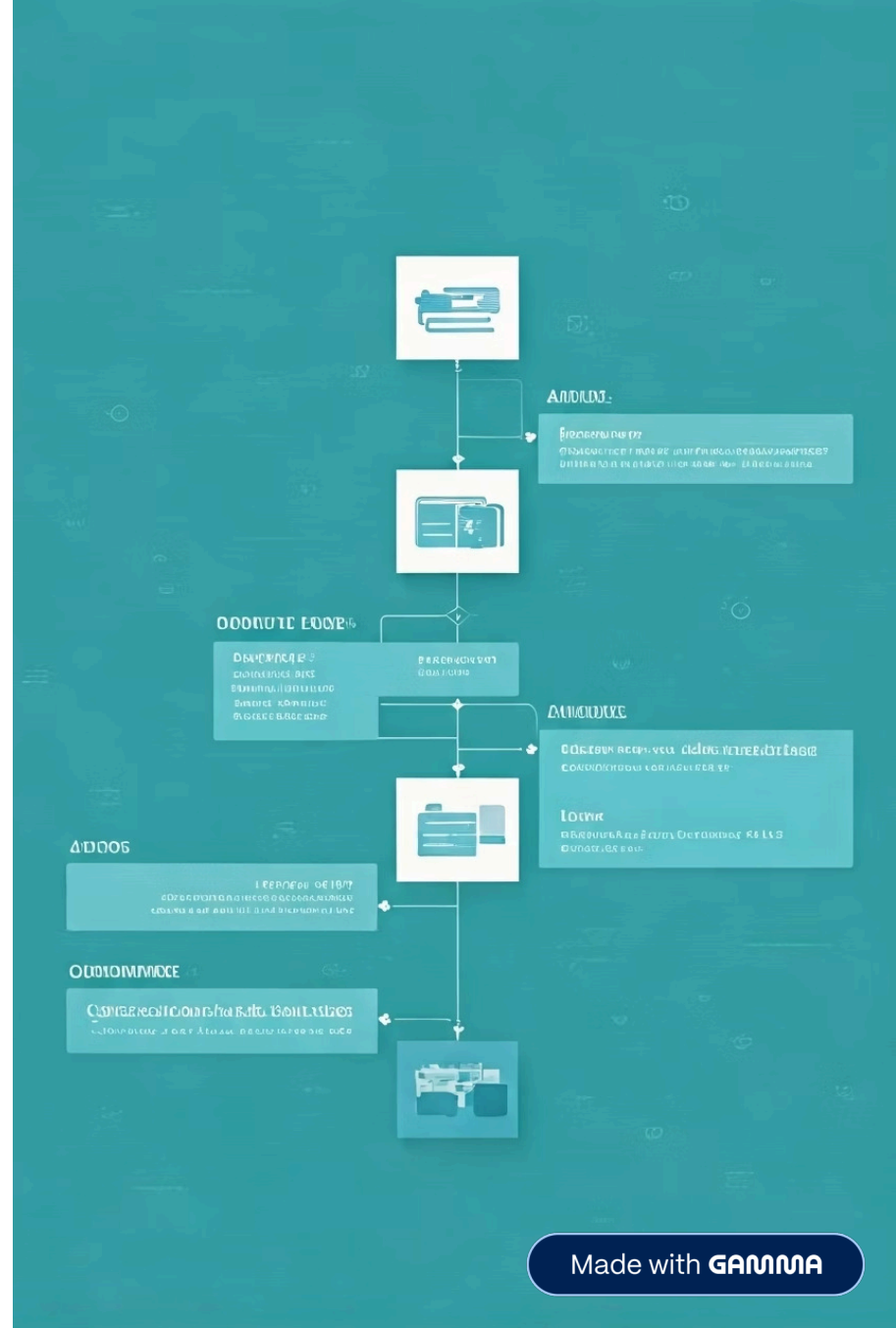
Large numeric datasets

Efficient memory usage

Lists

General-purpose collections

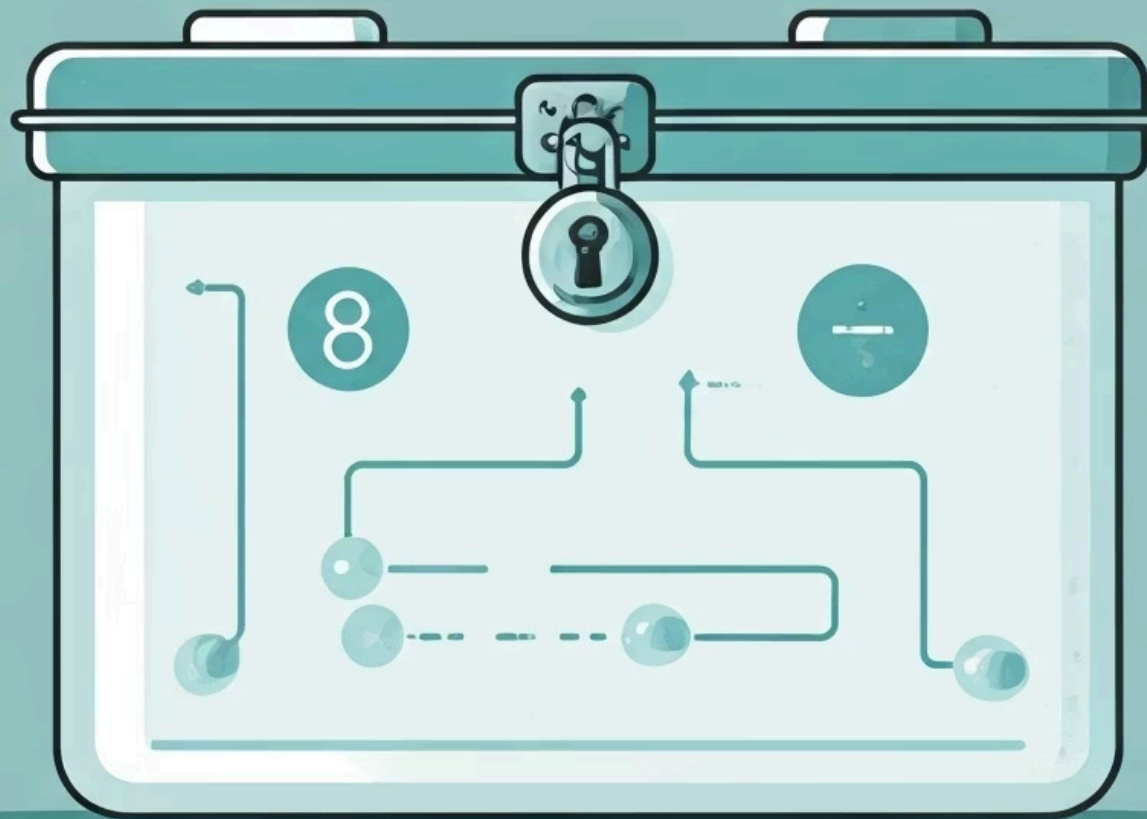
Mixed types, more methods



Introduction to Tuples

Immutable (unchangeable) sequences defined with parentheses

(1, 2, 3) - once created, cannot be changed



Tuple Operations & Examples

01

Access

t[0] gets first element

02

Unpack

$$x, y = (10, 20)$$

03

Convert

tuple([1, 2, 3])

Quick Revision: Array vs List vs Tuple

Feature	Array	List	Tuple
Mutability	Mutable	Mutable	Immutable
Data types	Homogeneous	Heterogeneous	Heterogeneous
Use case	Numeric data	General use	Fixed data



Practice Questions

1. Create an array of integers and append a number
2. Make a list with mixed types, remove an element
3. Create and unpack a tuple into variables
4. Why are tuples immutable? Give example use
5. Count occurrences of a number in an array

Summary & Next Steps

Arrays

Efficient, fixed-type

Lists

Flexible, dynamic

Tuples

Immutable, fixed

Practice coding these structures to master them

Explore: [GeeksforGeeks](#), [w3Schools](#)

