Unitedworld Institute of Technology # Data Science

Contents

❖ Data Types

❖ Arrays

❖ Library : Numpy

By:
Dr. Shikha Deep

Data types are the classification or categorization of data items. It represents the kind of value a variable holds.

- Zelle, J. (2016). Python Programming: An Introduction to Computer Science (3rd ed.)

Data types tell Python what kind of data a variable holds - **numbers, text, boolean**, etc. Knowing the type helps Python decide what operations can be performed.

Type	Description	Example
int	Whole numbers	x = 10
float	Decimal numbers	y = 3.14
str	Text (string)	name = "Shikha"
bool	Boolean (True, False)	status = True
complex	Complex numbers (real + imag)	z = 2 + 3j

KARNAVATI UNIVERSITY **Unitedworld Institute of Technology**

NOTE: type(value)

Code:

```
print(type(5))
```

print(type([1, 2, 3]))

print(type({"a": 1}))

Output:

<class 'int'>

<class 'list'>

class 'dict'>

<u>Array</u>

An array is a collection of elements, all of the same type, arranged in a contiguous block of memory and accessed by an index.

- R.L. Kruse, Data Structures and Program Design in C

- In Python, an array is a data structure that stores multiple values of the same data type in a single variable.

- Like – integers, decimal, complex, strings, Boolean etc.

Unitedworld Institute of Technology When to Use Array in Python?

- When you're dealing with many numbers of same type (e.g., integers or floats)

- When you want better performance and lower memory usage

- When doing scientific computing, we use NumPy

Python Array using array module

```
import array
# create an array of integers
marks = array.array('i', [80, 85, 90, 95])

# access elements
print(marks[0])
print(marks[2])
```

OUTPUT:

Format

array.array(typecode, [elements])

Typecode	Meaning
'i'	Integer
'f'	Float
'u'	Unicode char (rare)

Use NumPy Arrays (Instead of Lists):

Feature	Python List	NumPy Array (np.array)
Data type consistency	Can hold mixed types	All elements must be of same type
Performance	Slower for large data	Much faster for computation
Memory usage	Higher	Lower
Math operations	Not direct (for loops needed)	Vectorized (e.g., a + b works directly)
Used in libraries	Not used in ML/DS libraries	Used in pandas, sklearn, TensorFlow, etc.

Import numpy as np

EXAMPLES:

Example 1: Create numpy

```
import numpy as np
arr = np.array([85, 90, 78, 92])
print(arr)
```

Output:

[85 90 78 92]

Example 2: Add 5 marks to all students (Basic Operations)

print(arr + 5)

Output:

[90 95 83 97]

<u>Example 3: Mean, Max, Min, Shape</u>

print("Mean:", arr.mean())

print("Max:", arr.max())

print("Min:", arr.min())

print("Shape:", arr.shape)

OUTPUT:

Mean: 86.25

Max: 92

Min: 78

Array Methods

|	
append()	Adds an element at the end of the list
clear()	Removes all the elements from the list
copy()	Returns a copy of the list
count()	Returns the number of elements with the specified value
extend()	Add the elements of a list (or any iterable), to the end of the current list
index()	Returns the index of the first element with the specified value
insert()	Adds an element at the specified position
pop()	Removes the element at the specified position
remove()	Removes the first item with the specified value

reverse()	Reverses the order of the list
sort()	Sorts the list

KARNAVATI
UNIVERSITY **Unitedworld Institute of Technology**

Task to do

- Create a NumPy array of 5 subject marks.
- Add 10 bonus marks to all using + operator.
- Find average and max using mean() and max().
- Convert a Python list to NumPy array.
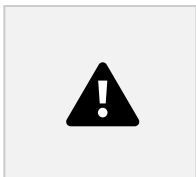
⚠ **Unitedworld Institute of Technology**

N-dimensional array

A dimension refers to the number of directions in which you can

index or access elements in an array.

Dimension	Meaning	Example
1D	A single list of items (like a line)	[1, 2, 3]
2D	A list of lists (like rows and columns)	[[1, 2], [3, 4]]
3D+	An array with layers (like a cube)	[[[1, 2], [3, 4]], [[5, 6], [7, 8]]]

⚠ **Unitedworld Institute of Technology** **1 D array:**

A 1D array is a linear collection of elements of the same data type, stored in contiguous memory locations.

2 D array:

A two-dimensional array (2D array) is a collection of elements arranged in rows and columns, forming a matrix-like structure.

3 D array:

A 3D array in Python is a data structure that organizes elements in three dimensions: rows, columns, and depth. It can be visualized ube or a stack of 2D matrices layered on top of each other.

⚠️

Examples:

Ex1:

import numpy as np

<mark>arr1d = np.array([10, 20, 30, 40])</mark>

print(arr1d)

print("Shape:", arr1d.shape)

```
print("Dimensions:", arr1d.ndim)
```

OUTPUT:

[10 20 30 40]

Shape: (4,)

Dimensions: 1

⚠️ **Unitedworld Institute of Technology Ex2:**

Table of marks of 3 students in 3 subjects:

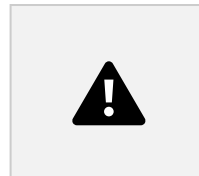[[85, 90, 88], ← Student 1

  [78, 80, 85], ← Student 2

  [92, 94, 96]] ← Student 3

CODE:	OUTPUT

marks = np.array([
    [85, 90, 88],
    [78, 80, 85],
    [92, 94, 96]
    ])
print("2D Array - Student
Marks:") print(marks)
print("Shape:", marks.shape)
	2D Array - Student
	Marks: [[85 90 88]
	[78 80 85]
	[92 94 96]]
	
	Shape: (3, 3)

EX 3: Seating Chart in a Classroom

Imagine 2 rows and 3 columns of seats labeled as seat numbers.

```
seats = np.array([
 [101, 102, 103], # Row 1
 [104, 105, 106] # Row 2
```

```
])
```

print("Seat:", seats[1][2])

OUTPUT:

?? 2nd Row, 3rd seat

General view of 3D

Shape = (layers, rows, columns)
= (depth, height, width)
= (semester, student, subject)

Ex4 : Student Marks in 2 Semesters

(3 Students × 2 Subjects)

Semester	Student	Subjects (Math, English)
1	3 rows	2 columns
2	3 rows	2 columns

import numpy as np

```
marks = np.array([
 [[85, 90], [78, 82], [92, 88]], # Semester 1
 [[86, 91], [80, 83], [94, 89]] # Semester 2
])
```

```
print("Shape:", marks.shape)
print("Marks of Student 2 in Semester 2, Subject 1:", marks[1][1][0])
```

OUTPUT:

Shape: (2, 3, 2)

⚠ s of Student 2 in Semester 2, Subject 1: 80

NumPy

- NumPy is the fundamental package for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays.

- Harris et al. (2020), Nature: Array Programming with NumPy.

- **NumPy** stands for **Numerical Python**.
- It gives us fast, powerful **arrays**, **math functions**, and **matrix operations** — all faster than regular Python lists.

Unitedworld Institute of Technology Why Use NumPy?

Feature	Benefit
Fast array operations	Much faster than Python lists
Built-in math/statistics	mean(), sum(), std()
Works with ML/Data Science	Required by pandas, sklearn, TensorFlow, etc.

Handles matrices and vectors	Easily supports dot product, transpose, etc.
Less memory	More compact storage than lists

⚠️

How to Use NumPy?

Step 1: Install

pip install numpy

Step 2: Import in Python

import numpy as np

Unitedworld Institute of Technology