



TECHNOCATION FREELANCING TRAINING INSTITUTE & SOFTWARE HOUSE

Professional NumPy Course Outline

Module 1: Introduction to NumPy

- What is NumPy? Why is it Important?
- Installing and Setting Up NumPy (`pip install numpy`)
- Understanding NumPy Arrays vs. Python Lists
- Creating and Initializing NumPy Arrays (`np.array()`, `np.zeros()`, `np.ones()`, `np.linspace()`)

Module 2: NumPy Array Operations

- Indexing, Slicing, and Iterating Over Arrays
- Reshaping and Resizing Arrays
- Stacking and Splitting Arrays
- Copying vs. Viewing Arrays (`.copy()`, `.view()`)

Module 3: Mathematical and Statistical Functions

- Element-wise Arithmetic Operations (+, -, *, /, %)
- Broadcasting in NumPy
- Aggregation Functions (`sum()`, `mean()`, `median()`, `std()`, `var()`)
- Trigonometric Functions (`sin()`, `cos()`, `tan()`)
- Exponential and Logarithmic Functions

Module 4: Advanced Array Manipulation

- Filtering and Boolean Indexing
- Sorting Arrays (`np.sort()`, `argsort()`)
- Searching Arrays (`where()`, `nonzero()`, `argmax()`, `argmin()`)
- Clipping and Rounding Functions

Module 5: Working with Multidimensional Arrays

- Creating and Manipulating 2D and 3D Arrays
- Transposing and Swapping Axes (`.T`, `swapaxes()`)
- Flattening and Raveling Arrays
- Applying Functions Along Axes (`apply_along_axis()`)

Module 6: Linear Algebra with NumPy

- Matrix Operations (`dot()`, `matmul()`, `inner()`, `outer()`)
- Eigenvalues and Eigenvectors (`np.linalg.eig()`)
- Solving Linear Equations (`np.linalg.solve()`)
- Determinant and Inverse of a Matrix (`np.linalg.det()`, `np.linalg.inv()`)

Module 7: Random Number Generation with NumPy

- Generating Random Integers and Floats (`rand()`, `randn()`, `randint()`)
- Setting Random Seeds for Reproducibility
- Sampling from Distributions (Normal, Binomial, Poisson)
- Shuffling and Permutation of Arrays

Module 8: Performance Optimization with NumPy

- Understanding NumPy's Memory Efficiency
- Vectorization vs. Looping in Python
- Using `numba` and `cython` for Speed Optimization
- Benchmarking NumPy Performance with `%timeit`

Final Module: Capstone Project & Certification

- Implementing a Real-World Data Processing Task Using NumPy
- Working with Large Datasets and NumPy Performance Optimization
- Final Presentation and Course Completion Certification