Topic	Sub Topic	Explanation
Day 1: Python Basics	1. Introduction to Python	What is Python? Why Python?
		Setting up Python (Installation & IDEs: Jupyter Notebook, VS Code, or PyCharm)
		Running Python scripts vs. Interactive mode
	2. Basic Syntax & Data Types	Variables and Data Types (int, float, str, bool)
		Taking user input (input())
		Type conversion (int(), float(), str())
	3. Operators	Arithmetic (+, -, *, /, //, %, **)
		Comparison (==, !=, >, <, >=, <=)
		Logical (and, or, not)
		Assignment (+=, -=, *=, /=, etc.)
	4. Conditional Statements	if, elif, else
		Nested conditions
Day 2: Loops, Functions & Strings	1. Loops	for loop
		while loop
		break, continue, pass
	2. Functions	Defining functions (def)
		Parameters and return values
		Default and keyword arguments
	3. String Manipulation	String indexing and slicing
		String methods (upper(), lower(), strip(), split(), replace(), etc.)
		f-strings and format()
		Working with multi-line strings

Day 3: Data Structures in Python	1. Lists	Creating lists, indexing, slicing
		List methods (append(), remove(), pop(), sort(), reverse(), etc.)
		List comprehension
	2. Tuples	Tuple properties (immutable)
		Tuple unpacking
		3. Dictionaries
		Key-value pairs
		Dictionary methods (keys(), values(), items(), get(), update(), etc.)
	4. Sets	Set operations (union, intersection, difference)
	5. Working with Files	Reading and writing files (open(), read(), write(), with statement)
Day 4: Error Handling, Regular Expressions & OOP	1. Error & Exception Handling	try, except, finally
		Handling specific exceptions (ZeroDivisionError, ValueError, etc.)
		Raising exceptions (raise)
	2. Regular Expressions (re module)	findall(), search(), match()
		Using regex patterns (\d, \w, \s, +, *, {}, |)
	3. Object-Oriented Programming (OOP)	Class and objects
		Constructor (__init__)
		Instance and class variables
		Inheritance
		Arrays vs. Lists

Day 5: Advanced Libraries	1. NumPy	Creating NumPy arrays (array(), arange(), linspace())
		Indexing and slicing
		Operations on arrays (mathematical and statistical functions)
	2. Pandas	Series and DataFrames
		Reading and writing CSV files
		DataFrame operations (head(), tail(), describe(), info(), groupby(), merge())
		Handling missing values
	3. Matplotlib & Seaborn (Visualization)	Line plots, bar plots, scatter plots
		Histograms, box plots
		Customizing plots (labels, titles, legends)
Day 6: Machine Learning Project	Project: Linear Regression OR Classification	Dataset: Use a dataset from sklearn.datasets or a CSV file
		Steps:
		Load dataset using Pandas
		Preprocess data (handle missing values, categorical variables)
		Split data into training and testing sets (train_test_split)
		Train a model using LinearRegression (for regression) or LogisticRegression (for classification)
		Evaluate model performance using metrics like R^2 score, accuracy, precision-recall
		Visualize results using Matplotlib/Seaborn