Training Plan: Python From Scratch - Weekly 1-Hour Sessions

Overview: This document outlines a 12-week corporate training plan to teach Python from scratch. Each session will be 1 hour, with focus on theory, live demos, and exercises. The deliverable will include a weekly PDF (converted from PowerPoint).

Weeks:

---

**Week 1: Python Introduction and Basics** - Evolution of Python - Version history and differences (Python 2 vs Python 3) - Installing Python: global vs virtual environment - Features of Python - Different ways to run a Python program (REPL, script, IDE, notebooks) - IDEs and frameworks supported (VS Code, PyCharm, Jupyter, Flask, Django, etc.) - Python keywords and identifiers - Writing comments in Python - Data types overview - Variables and naming conventions - Introduction to functions - Overview of control flow: if, for, while (to be detailed later) - Exercise: Setup environment, Hello World, basic I/O

**Week 2: Variables, Data Types, and Operators** - Detailed Python data types: int, float, str, bool, None - Type conversion and type() function - Operators: arithmetic, assignment, comparison, logical, bitwise, membership, identity - Mutability and immutability - Dynamic typing in Python - Exercise: Input conversion, basic calculator, type identification

**Week 3: Strings and String Manipulation** - Creating and accessing strings - String methods: upper(), lower(), find(), replace(), split(), join() - Slicing and indexing - String formatting (%, format(), f-strings) - Escape characters and raw strings - Multi-line strings and docstrings - Exercise: Format messages, reverse string, count vowels

**Week 4: Data Structures - Lists, Tuples, Sets, Dictionaries** - List creation, indexing, slicing, methods - Tuple characteristics and usage - Set operations: union, intersection, difference - Dictionary key-value manipulation - Comprehensions (list, set, dict) - Exercise: Create contact book, item frequency counter

**Week 5: Flow Control** - if, elif, else blocks - Nested conditions - Boolean expressions and logical operators - while and for loops - range() usage - Loop control: break, continue, pass - Exercise: FizzBuzz, factorial, pattern printing

**Week 6: Functions and Scopes** - def and return - Positional, keyword, default arguments - Variable-length arguments (*args,* *kwargs) - Local vs global scope - Recursion basics - Lambda functions - Exercise: Create a calculator, Fibonacci series, recursion demo

**Week 7: Modules, Packages, and File Handling** - import, from-import, aliasing - Creating and using custom modules - Installing packages via pip - File operations: open, read, write, append - File modes and context manager (with) - Working with CSV and JSON files - Exercise: Read/write file, JSON formatter

**Week 8: Exception Handling and Debugging** - try, except, else, finally - Catching multiple exceptions - Raising exceptions - Built-in exceptions - Debugging tips and using IDE features - Exercise: Robust user input, debug trace

**Week 9: Object-Oriented Programming - Part 1** - Classes and objects - **init** constructor - Instance vs class variables - Instance methods - Magic methods (**str**, **repr**) - Exercise: Employee class, product inventory

**Week 10: Object-Oriented Programming - Part 2** - Inheritance and method overriding - super() function - Encapsulation and access modifiers - Class methods and static methods - Polymorphism - Exercise: Banking system with inheritance

**Week 11: Python Standard Libraries and Useful Modules** - os, sys, datetime, math, random - json, shutil, pathlib - Working with environment variables - Exploring built-in help and documentation - Exercise: Directory cleanup tool, random password generator

**Week 12: Capstone & Real-World Applications** - Overview of popular frameworks: Flask, Django, pandas, numpy, etc. - Automating tasks with Python - Mini-project ideas: - To-do app - File organizer - Simple REST API using Flask - Wrap-up and revision - Q&A and feedback

---

Next Step: Weekly PDF slide decks will be prepared based on this structure. Each PDF will contain: - Topic breakdown - Visual examples - Demo code - 1-2 exercises