Python Topics — Revision Checklist

🔷 **Level 1 — Basics (Foundation)**

- Variables, Data types (int, float, str, bool, None)
- Type conversion (casting)
- Operators (arithmetic, comparison, logical, assignment)
- String manipulation — f-strings, slicing, .split(), .join(), .strip(), .format()
- Input/Output (input(), print())
- Conditional statements (if/elif/else)
- Loops (for, while), break/continue/pass
- range(), nested loops

🔷 **Level 2 — Data Structures**

- **List** — indexing, slicing, list methods, list comprehension
- **Tuple** — immutability, unpacking
- **Dictionary** — key-value operations, .get(), .items(), dict comprehension
- **Set** — union, intersection, use-cases
- Nested data structures (list of dicts, dict of lists — এগুলো তুমি প্রচুর দেখবে JSON/API response handle করার সময়)

🔷 **Level 3 — Functions**

- Function definition, parameters, return values
- Default arguments, keyword arguments, *args, **kwargs
- Lambda functions
- Scope (local vs global)
- **Docstrings** — কারণ MCP tools-এ docstring থেকেই LLM বুঝে tool কী করে
- Recursion (basic level)

🔷 **Level 4 — Object-Oriented Programming (OOP) — খুবই গুরুত্বপূর্ণ**

- Class, object, __init__
- Instance vs class variables
- Methods (instance, class, static)
- Inheritance, Method overriding
- super()
- Dunder/magic methods (__str__, __repr__, __call__)
- Encapsulation basics (public/private convention)

কেন গুরুত্বপূর্ণ: LangChain, CrewAI, MCP — সব framework-ই class-based। Agent, Tool, Server — সবকিছু class দিয়ে define হয়।

🔷 **Level 5 — Error Handling**

- try/except/else/finally
- Multiple exceptions, custom exceptions (raise)
- Common exceptions (ValueError, KeyError, TypeError, FileNotFoundError)

🔷 **Level 6 — File & Data Handling**

- File read/write (open(), context manager with)
- Working with JSON (json.load, json.dumps) — **must-know**, কারণ API response সাধারণত JSON
- CSV handling (basic)
- Working with paths (os, pathlib)

🔷 **Level 7 — Modules & Packages**

- import, creating your own modules
- pip install, virtual environments (venv)
- requirements.txt
- Standard library essentials: os, sys, datetime, random, time

🔷 **Level 8 — Type Hints & Pydantic (Modern Python — MCP/Agent-এর জন্য critical)**

- Type hints (def foo(x: int) -> str:)
- Optional, Union, List, Dict from typing
- **Pydantic basics** — BaseModel, field validation
    - কেন: MCP tools, structured output, LangChain-এর data validation সব Pydantic দিয়ে হয়

🔷 **Level 9 — Decorators (Must-know for MCP)**

- কীভাবে decorator কাজ করে (@decorator_name)
- নিজে decorator লেখা
- Built-in decorators: @staticmethod, @classmethod, @property
    - কেন: FastMCP-এ @mcp.tool(), @mcp.resource() — সবকিছু decorator pattern

🔷 **Level 10 — Asynchronous Programming (Critical for Agent/MCP work)**

- async def, await
- asyncio basics — asyncio.run(), asyncio.sleep()
- Sync vs async — কখন কোনটা দরকার
- async with API calls (httpx, aiohttp — অন্তত concept level-এ বুঝলেই চলবে)

এটা সবচেয়ে বেশি লোকে skip করে, কিন্তু MCP server আর Agent framework-এর কোর প্রায় পুরোটাই async-based।

🔷 **Level 11 — Working with APIs**

- requests library — GET/POST, headers, params
- API response handle করা (status code, JSON parsing)
- Environment variables (.env, python-dotenv) — API key secure রাখার জন্য

🔷 **Level 12 — Iterators & Generators (হালকা জানলেই চলবে)**

- yield কীভাবে কাজ করে
- Generator expression vs list comprehension
- কেন দরকার: streaming responses (LLM output token-by-token আসে) বোঝার জন্য কাজে লাগবে

Priority Order (যদি সময় কম থাকে)

যদি সময় সীমিত থাকে, এই order-এ focus করো:

1. **Must (স্কিপ করা যাবে না):** OOP, Functions, Dictionaries/JSON, Error handling, Type hints
2. **High priority:** Decorators, Async/await, Pydantic basics, requests/API handling
3. **Nice to have:** Generators, advanced list comprehension, file handling deep dive

---

চাইলে প্রতিটা topic-এর জন্য একটা ছোট practice exercise বানিয়ে দিতে পারি, যাতে শুধু পড়ে না গিয়ে হাতে-কলমে test করতে পারো — বলো লাগবে কিনা।