Python Reference - Short Notes

1. **Library Functions:**

- Predefined functions in Python (math, random, os).

- Save time, improve code reusability.

2. **Keywords:**

- Reserved words with special meaning.

- Cannot be used as variable names. Eg: if, else, while.

3. **Inbuilt Functions:**

- print(), input(), len(), type(), int(), float(), str(), max(), min().

4. **Variables:**

- Containers to store data in memory.

- Dynamically typed, use id() and type().

5. **Data Types:**

- Numeric: int, float, complex.

- Boolean: bool.

- Collection: str, list, tuple, set, dict.

- Others: bytes, bytearray, memoryview, NoneType.

6. **List vs Tuple vs Set vs Dict:**

- List: Ordered, mutable.

- Tuple: Ordered, immutable.

- Set: Unordered, unique values.

- Dict: Key-value pairs.

7. **Data Manipulation Functions:**

- append(), insert(), pop(), add(), remove(), count(), index().

8. Indexing & Slicing:

- Indexing: Access elements by position.

- Slicing: Extract subsequence using start:stop:step.

9. Operators:

- Arithmetic: +, -, *, /, %, //, **.

- Logical: and, or, not.

- Bitwise: &, |, ^, ~, <<, >>.

- Relational: ==, !=, >, <, >=, <=.

- Assignment: =, +=, -=.

- Membership: in, not in.

- Identity: is, is not.

10. Indentation:

- Defines code blocks (4 spaces recommended).

11. Loops:

- For: Iterates over sequence.

- While: Repeats until condition false.

12. Functions:

- Types: Built-in, User-defined, Lambda, Recursive, Higher-order.

- Parameters: Positional, keyword, *args, **kwargs.

- Return values, local/global variables.

13. Comprehensions:

- Create collections in one line.

- Used for filtering and transformation.

14. Decorators, Iterators, Generators:

- Decorators: Modify functions.
- Iterators: Loop using next().
- Generators: Yield values lazily.

15. Modules & Packages:

- Modules: Reusable code files.
- Packages: Organized module collections.

16. OOP:

- Class & Object: Blueprint vs instance.
- Methods: Instance, class, static.
- Encapsulation: Public, protected, private.
- Abstraction: Hide details using ABC.
- Polymorphism: Same interface, different behavior.
- Inheritance: Single, multiple, multilevel, hierarchical, hybrid.