PYTHON QUESTIONS

1. **init keyword:**
    - Special method to initialize a class.
2. **self keyword:**
    - Refers to the instance of the class.
3. **lambda function:**
    - Short function: `lambda x: x + 1`
4. **lambda vs normal function:**
    - Lambda: One-line, no name.
    - Normal: Defined with `def`, can have many lines.
5. **Generators:**
    - Functions that yield values one by one.
    - Use when large data or memory-saving needed.

```
def gen():
    yield 1
    yield 2
```

6. **Python is interpreted:**
    - Code runs line by line, not compiled first.
7. **List vs Tuple:**
    - List: Can change.
    - Tuple: Cannot change.
8. **List vs Set:**
    - List: Keeps order, allows duplicates.
    - Set: No order, no duplicates.
9. **Use dictionary:**
    - When you want to map key to value.
10. **Decorators:**

- Change behavior of a function.

```
def deco(func):
    def wrapper():
        print("Before")
        func()
        print("After")
    return wrapper
```

11. **Iterators:**

- Objects you can loop through with `next()`.

12. **Slicing:**

- Getting parts of lists: `a[1:3]`

13. **Mutable vs Immutable:**

- Mutable: Can change (list).
- Immutable: Can't change (tuple).

14. **Python threads:**

- Supports multi-threading but limited by GIL.

15. **GIL (Global Interpreter Lock):**

- Allows only one thread to run Python code at once.

16. **Dislike in Python:**

- Slow speed, GIL limits threading.

17. **List Comprehension:**

- Short way to make lists: `[x for x in range(5)]`

18. **Dunder Methods:**

- Special methods: `__init__`, `__str__`, `__len__`

19. **init method:**

- Runs when object is created. Initializes attributes.

20. **Array vs Numpy:**

- Array: Basic.

- Numpy: Faster, supports more operations.

21. **Higher Order Functions:**

    - **Take other functions as input:** `map`, `filter`, `reduce`