Chapter 1: Python Basics

- **Topics**: Data types, variables, operators, conditionals (if, elif, else), and loops (for, while).
- **Project**: *System Health Checker* – A simple script to simulate CPU, memory, and disk health checks, printing a "healthy" or "alert" status based on threshold values.

1.1 Data Types

Data types are the kinds of values you can work with in Python.

- **Integer** (`int`): Represents whole numbers without a decimal. Examples include `10`, `-5`, and `0`.
- **Float** (`float`): Represents numbers with a decimal point, like `3.14` or `-0.001`.
- **String** (`str`): A sequence of characters, typically text, enclosed in quotes. Examples: `"Hello"`, `'SRE Script'`.
- **Boolean** (`bool`): Represents `True` or `False` values, used to control logic flow in code.

Example (`data_types.py`):

```
age = 30              # Integer
price = 19.99         # Float
name = "Alice"        # String
is_active = True      # Boolean

print("Age:", age)
print("Price:", price)
print("Name:", name)
print("Is Active:", is_active)
```

1.2 Variables

Variables act as containers for storing data values. They are created when you assign a value to a name using the `=` operator.

Rules:

- Variable names must start with a letter or an underscore, and can contain letters, numbers, and underscores.
- They are case-sensitive (`Name` and `name` are different variables).

Example (`variables.py`):

```
username = "SRE_Engineer"
max_retries = 5
timeout = 30.5

print("Username:", username)
print("Max Retries:", max_retries)
print("Timeout:", timeout)
```

1.3 Operators

Operators let you perform operations on variables and values.

Types of Operators:

1. **Arithmetic Operators**: Used for mathematical calculations.
   - `+` (addition), `-` (subtraction), `*` (multiplication), `/` (division), `%` (modulus), `**` (exponentiation), `//` (floor division).

   Example (`operators.py`):

   ```
   a = 10
   b = 3
   print("Addition:", a + b)
   print("Subtraction:", a - b)
   print("Multiplication:", a * b)
   print("Division:", a / b)
   print("Modulus:", a % b)
   print("Exponent:", a ** b)
   print("Floor Division:", a // b)
   ```

2. **Comparison Operators**: Compare two values and return `True` or `False`.
   - `==` (equal), `!=` (not equal), `>`, `<`, `>=`, `<=`.

   Example:

   ```
   is_equal = (10 == 10)  # True
   is_greater = (5 > 2)  # True
   print("Is Equal:", is_equal)
   print("Is Greater:", is_greater)
   ```

3. **Logical Operators**: Combine multiple conditions.
   - `and`: True if both conditions are true.
   - `or`: True if at least one condition is true.
   - `not`: Inverts the result.

   Example:

```python
is_within_range = (5 > 2) and (5 < 10)
print("Is within range:", is_within_range)
```

1.4 Conditionals (If Statements)

Conditionals allow you to execute specific code blocks based on conditions.

Syntax:

```python
if condition:
    # Code to execute if condition is true
elif another_condition:
    # Code to execute if the second condition is true
else:
    # Code to execute if no conditions above are true
```

Example (`conditionals.py`):

```python
threshold = 70
usage = 85

if usage > threshold:
    print("Alert: High usage!")
else:
    print("Usage is within limits.")
```

1.5 Loops

Loops allow you to repeat a set of instructions.

1. **For Loop**: Used to iterate over a sequence (like a list, tuple, or range). **Example** (`loops.py`):

    ```python
    for i in range(3):  # Loops 3 times (i = 0, 1, 2)
        print("Iteration", i)
    ```

2. **While Loop**: Repeats as long as a condition is `True`.

    Example:

    ```python
    count = 0
    while count < 3:
    ```

```
        print("Count:", count)
        count += 1
```

Project: System Health Checker

Objective: Write a script to check system resource usage (CPU, memory, and disk) and print a warning if any resource exceeds a set threshold.

Project Code (`system_health_checker.py`):

```
# Define current usage values (simulated)
cpu_usage = 65
memory_usage = 70
disk_usage = 80

# Define threshold limits
cpu_threshold = 75
memory_threshold = 75
disk_threshold = 85

# Check CPU usage
if cpu_usage > cpu_threshold:
    print("Warning: High CPU usage!")
else:
    print("CPU usage is normal.")

# Check memory usage
if memory_usage > memory_threshold:
    print("Warning: High Memory usage!")
else:
    print("Memory usage is normal.")

# Check disk usage
if disk_usage > disk_threshold:
    print("Warning: High Disk usage!")
else:
    print("Disk usage is normal.")
```

Explanation:

- The script checks three system resources (CPU, memory, and disk).
- It compares each resource's usage against a predefined threshold.
- If any usage exceeds the threshold, a warning message is displayed. Otherwise, it reports that the usage is normal.

This project will help reinforce your understanding of **variables, conditionals, and basic output statements**. Let me know if you'd like more insights on modifying or expanding this script!