Introduction To Python

Syntax

- **Definition**: Syntax is the set of rules that define the correct structure and format of Python code. It includes things like indentation, line breaks, and keyword usage.
- **Sub-Areas**:
  - **Indentation**: Python relies on indentation to define code blocks, like loops or functions.
  - **Keywords**: Reserved words that have special meaning (e.g., `if`, `else`, `for`, `def`).
- **Example**:

```
if 5 > 2:
    print("5 is greater than 2")  # Indentation is required here
```

2. Comments

- **Definition**: Comments are lines of text in code that Python ignores when executing. They're used to add explanations or notes within code.
- **Sub-Areas**:
  - **Single-Line Comment**: Starts with `#`.
  - **Multi-Line Comment**: Usually enclosed in triple quotes `""" ... """`.
- **Example**:

```
# This is a single-line comment
"""
This is a multi-line comment
explaining multiple things.
"""
print("Comments are useful!")
```

3. Variables

- **Definition**: Variables are containers for storing data values. They can store different data types and can be reassigned.
- **Sub-Areas**:
  - **Variable Naming**: Names can include letters, numbers, and underscores but cannot start with a number.
  - **Assignment**: Using the `=` sign to assign a value to a variable.
- **Example**:

```
name = "Alice"  # Assigns the value "Alice" to the variable name
age = 30        # Assigns the value 30 to the variable age
```

4. Data Types

- **Definition**: Data types specify the kind of value a variable can hold, like a number, string, or Boolean.
- **Sub-Areas**:
  - **Integers**: Whole numbers, e.g., `10`, `-5`.
  - **Floats**: Decimal numbers, e.g., `3.14`, `-0.5`.
  - **Strings**: Text data enclosed in quotes, e.g., `"hello"`.
  - **Booleans**: Logical values, `True` or `False`.
- **Example**:

```
my_int = 10         # Integer
my_float = 3.14     # Float
my_string = "Hello" # String
my_bool = True      # Boolean
```

5. Numbers

- **Definition**: Numbers are data types used for mathematical operations.
- **Sub-Areas**:
  - **Integers**: Whole numbers.
  - **Floats**: Decimal-point numbers.
- **Example**:

```
x = 10     # Integer
y = 5.5    # Float
result = x + y  # result is 15.5
```

6. Strings

- **Definition**: Strings are sequences of characters, used for text data, enclosed in single or double quotes.
- **Sub-Areas**:
  - **Concatenation**: Combining strings using `+`.
  - **String Methods**: Built-in functions, like `.upper()`, `.lower()`, `.replace()`.
- **Example**:

```
first_name = "Alice"
last_name = "Smith"
full_name = first_name + " " + last_name  # Concatenates the names
print(full_name.upper())  # Prints "ALICE SMITH"
```

7. Booleans

- **Definition**: Booleans represent truth values `True` and `False`.
- **Sub-Areas**:
  - **Comparisons**: Evaluating conditions (e.g., ==, >, <).
  - **Logical Operations**: Using `and`, `or`, `not` with Boolean values.
- **Example**:

```
is_raining = False
is_warm = True
is_nice_weather = not is_raining and is_warm  # Combines using logical
operators
```

8. Operators

- **Definition**: Operators are symbols or keywords that perform operations on values and variables.
- **Sub-Areas**:
  - **Arithmetic Operators**: `+`, `-`, `*`, `/` for math.
  - **Assignment Operators**: `=`, `+=`, `-=`, etc., for assigning and updating values.
  - **Comparison Operators**: `==`, `!=`, `>`, `<` for comparisons.
  - **Logical Operators**: `and`, `or`, `not` for Boolean logic.
- **Example**:

```
a = 10
b = 20
print(a + b)  # Arithmetic addition
print(a == b)  # Comparison
```

Syntax

- Syntax is the set of rules that define the structure of Python code. It specifies how code statements are constructed to execute properly.

Example:

```
# Simple print statement following Python syntax
print("Hello, Python!")
```

2. Comments

- Comments are lines in the code that Python ignores. They're used to add notes and explanations for readers of the code.

Example:

```
# This is a single-line comment
print("Comments are not executed")
```

3. Variables

- Variables store data values. In Python, you don't need to declare the type of variable; it's dynamically assigned based on the value.

Example:

```
age = 25
name = "Alice"
print(name, "is", age, "years old.")
```

4. Data Types

- Python has several built-in data types, such as integers, floating-point numbers, strings, and Booleans.

Examples:

```
# Integer
age = 30

# Float
price = 19.99

# String
name = "Bob"

# Boolean
is_student = True
```

5. Numbers

- Numbers in Python include integers, floats, and complex numbers.

Examples:

```
x = 10       # integer
y = 3.14     # float
z = x + y    # addition of integer and float results in float
print(z)
```

6. Strings

- Strings represent sequences of characters and are defined with quotes.

Examples:

```
greeting = "Hello"
name = "World"
message = greeting + ", " + name + "!"
print(message)
```

7. Booleans

- Boolean values are `True` or `False`. They're often used in conditions and comparisons.

Examples:

```
is_active = True
print(is_active)
```

8. Operators

- Python supports several types of operators, including arithmetic, assignment, comparison, logical, and more.

Examples:

```
# Arithmetic operators
sum = 5 + 3
diff = 10 - 4
product = 7 * 3
quotient = 8 / 2

# Comparison operators
is_equal = (5 == 5)
is_greater = (7 > 3)

# Logical operators
and_op = (True and False)  # results in False
```

```
or_op = (True or False)    # results in True
```

Lab Script: Python Basics

Here’s a lab script that uses all these concepts. You can run this code to practice.

```
# Syntax and Comments
# This is a comment explaining what the lab does
print("Python Basics Lab")  # Outputs a simple message

# Variables and Data Types
name = "Alice"               # String data type
age = 25                     # Integer data type
height = 5.7                 # Float data type
is_student = True            # Boolean data type

print(name, "is", age, "years old and", height, "feet tall.")  # Example of
variable usage

# Numbers
a = 10
b = 4
sum_ab = a + b
difference_ab = a - b
product_ab = a * b
quotient_ab = a / b
print("Sum:", sum_ab, "Difference:", difference_ab, "Product:", product_ab,
"Quotient:", quotient_ab)

# Strings
greeting = "Hello"
message = greeting + ", " + name + "!"
print(message)

# Booleans
is_tall = height > 6
print("Is tall:", is_tall)

# Operators
is_adult = age >= 18
is_child = age < 18
print("Is an adult:", is_adult)
print("Is a child:", is_child)

# Logical Operators
can_register = is_adult and is_student
print("Can register:", can_register)

# End of lab script
```