2. Student Handout

Introduction to Python and Basic Syntax

Welcome to your first lesson on Python! This handout will guide you through the basics of Python, including its syntax, variables, data types, and basic operators. Let's get started!

---

1. Overview of Python

History of Python

- Created by **Guido van Rossum** and first released in **1991**.
- Named after the British comedy group **Monty Python**.

Features of Python

- **Easy to Learn and Use**: Simple syntax resembling plain English.
- **Interpreted Language**: Executes code line by line.
- **Cross-Platform**: Works on Windows, macOS, and Linux.
- **Extensive Libraries**: Supports web development, data analysis, machine learning, etc.
- **Community Support**: Large and active community for help and resources.

Applications of Python

- **Web Development**: Using frameworks like Django and Flask.
- **Data Science and Machine Learning**: Libraries like Pandas, NumPy, and TensorFlow.
- **Automation**: Automating repetitive tasks with scripts.
- **Game Development**: Creating games with libraries like Pygame.
- **Scripting**: Writing small scripts for automation.

---

2. Setting Up the Python Environment

Installation

1. **Download Python**: Visit python.org and download the latest version.
2. **Install Python**: Follow installation instructions and check "Add Python to PATH".

IDE Setup

- **PyCharm**: A powerful IDE with many features.
- **VS Code**: A lightweight editor with Python support.
- **Jupyter Notebook**: Ideal for data science and interactive coding.

---

3. Understanding Python Syntax and Indentation

What is Syntax?

- Syntax refers to the rules for writing code.

Python Syntax

- Example 1:

  ```python
  print("Hello, World!")
  ```

- Example 2:

  ```python
  print("Python is fun!")
  ```

- Example 3:

  ```python
  print("Let's learn Python.")
  ```

Indentation in Python

- Indentation defines code blocks.

- Example 1:

  ```python
  if 5 > 3:
      print("5 is greater than 3")
  ```

- Example 2:

  ```python
  for i in range(5):
      print(i)
  ```

- Example 3:

  ```python
  while True:
      break
  ```

---

4. Variables and Data Types

What is a Variable?

- A variable stores data.

Data Types in Python

1. **Integers**: Whole numbers.
   - Example 1: `x = 5`
   - Example 2: `age = 30`
   - Example 3: `count = -10`
2. **Floats**: Decimal numbers.
   - Example 1: `y = 3.14`
   - Example 2: `temperature = 98.6`
   - Example 3: `price = -0.99`
3. **Strings**: Text.
   - Example 1: `name = "Python"`
   - Example 2: `greeting = "Hello"`
   - Example 3: `city = "New York"`

4. **Booleans**: True or False values.
    - Example 1: `is_active = True`
    - Example 2: `is_valid = False`
    - Example 3: `is_open = True`

---

5. Basic Operators

Arithmetic Operators

- **Addition**: `+`
    - Example 1: `5 + 3` results in `8`
    - Example 2: `10 + 20` results in `30`
    - Example 3: `-5 + 5` results in `0`
- **Subtraction**: `-`
    - Example 1: `5 - 3` results in `2`
    - Example 2: `20 - 10` results in `10`
    - Example 3: `-5 - 5` results in `-10`
- **Multiplication**: `*`
    - Example 1: `5 * 3` results in `15`
    - Example 2: `10 * 2` results in `20`
    - Example 3: `-5 * 5` results in `-25`
- **Division**: `/`
    - Example 1: `5 / 3` results in `1.6667`
    - Example 2: `10 / 2` results in `5`
    - Example 3: `-10 / 5` results in `-2`

Comparison Operators

- **Equal to**: `==`
    - Example 1: `5 == 5` is `True`
    - Example 2: `10 == 5` is `False`
    - Example 3: `-5 == -5` is `True`
- **Not equal to**: `!=`
    - Example 1: `5 != 3` is `True`
    - Example 2: `10 != 10` is `False`

    - Example 3: `-5 != 5` is `True`
- **Greater than**: `>`
    - Example 1: `5 > 3` is `True`
    - Example 2: `10 > 20` is `False`
    - Example 3: `-5 > -10` is `True`
- **Less than**: `<`
    - Example 1: `5 < 3` is `False`
    - Example 2: `10 < 20` is `True`
    - Example 3: `-5 < -10` is `False`

Logical Operators

- **and**: Both conditions must be true.
    - Example 1: `(5 > 3) and (3 > 1)` is `True`
    - Example 2: `(5 > 3) and (3 < 1)` is `False`
    - Example 3: `(5 < 3) and (3 > 1)` is `False`
- **or**: At least one condition must be true.
    - Example 1: `(5 > 3) or (3 < 1)` is `True`
    - Example 2: `(5 < 3) or (3 > 1)` is `True`
    - Example 3: `(5 < 3) or (3 < 1)` is `False`
- **not**: Reverses the condition.
    - Example 1: `not (5 > 3)` is `False`
    - Example 2: `not (5 < 3)` is `True`
    - Example 3: `not (3 == 3)` is `False`

---

6. Writing and Running Your First Python Program

Example Program

```
name = input("What is your name? ")
print("Hello, " + name + "!")
```

Activity: Write a Simple Python Script

Write a Python script that calculates the sum of two numbers entered by the user.

```python
num1 = input("Enter the first number: ")
num2 = input("Enter the second number: ")
sum = int(num1) + int(num2)
print("The sum is: " + str(sum))
```

Conclusion

You've completed your first lesson on Python! We covered:

- The history and features of Python.
- Setting up the Python environment.
- Basic Python syntax and indentation.
- Variables and data types.
- Basic operators.
- Writing and running your first Python program.

Keep practicing, and soon you'll be writing Python code like a pro!

Next Steps

In the next lesson, we'll explore **control structures** like loops and conditionals. Happy coding! 😊