
🎓 Lecture: Python Syntax for Beginners

🎤 Introduction

Assalamu Alaikum!
Welcome to *Jamia Pakistan*, your gateway to learning computers, AI, programming, and much more!

In today's lecture, we are going to start our **Python journey** with the most fundamental concept — **Syntax**.

Just like every language has grammar, Python has **syntax** — a set of rules that tells the computer how to read and execute your code.

🔹 1. What is Syntax in Python?

Definition:
Syntax in Python refers to the rules that define how you write Python code so that it runs properly without errors.

Think of syntax as the **grammar** of the Python language.
If you don't follow these rules, Python will give you an error.

🔸 2. Case Sensitivity

Python is **case-sensitive**.

```
Name = "Azhar"
name = "Hussain"
```

These two are **different variables**.
So be careful when naming your variables!

🔹 3. Indentation

Unlike other programming languages that use `{ }` or keywords like `begin/end`, Python uses **indentation** to define blocks of code.

✅ Correct

```
if True:
    print("Hello, Jamia Pakistan!")
```

❌ Incorrect

```
if True:
print("Hello, Jamia Pakistan!")  # This will give an error
```

➡ **Always use 4 spaces** (or a tab) for indentation.

🔸 4. Comments

Comments are used to explain your code. Python ignores them when running the program.

- **Single-line comment:**

```
# This is a comment
```

- **Multi-line comment:**

```
"""
This is a
multi-line comment
"""
```

Use comments to make your code more understandable!

🔹 5. Variables and Data Types

In Python, you don't need to mention the data type when creating a variable.

```
name = "Azhar"      # String
age = 24            # Integer
pi = 3.14           # Float
is_active = True    # Boolean
```

Python will automatically understand the type of data.

🔸 6. Print Statement

Use `print()` to display output:

```
print("Welcome to Jamia Pakistan!")
```

🔹 7. Taking Input

To take input from the user:

```
name = input("Enter your name: ")
print("Hello", name)
```

🔸 8. Conditional Statements

Use `if`, `elif`, and `else` to make decisions in your code.

```
age = 18

if age >= 18:
    print("You can vote!")
```

```
else:
    print("You are too young to vote.")
```

🔹 9. Loops

Python supports two main types of loops:

For Loop:

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

While Loop:

```
i = 0
while i < 5:
    print(i)
    i += 1
```

🔸 10. Functions

A function is a block of code that performs a specific task.

```
def greet(name):
    print("Hello", name)

greet("Azhar")
```

🔹 11. Lists and Dictionaries

List:

```
fruits = ["apple", "banana", "mango"]
print(fruits[0])
```

Dictionary:

```
student = {"name": "Azhar", "age": 24}
print(student["name"])
```

🔸 12. Error Handling

Use try-except to catch errors and prevent your program from crashing.

```
try:
    print(10 / 0)
except ZeroDivisionError:
    print("You can't divide by zero!")
```

📌 Conclusion

Congratulations! You now understand the basic **syntax of Python**.

Let’s recap what we covered:

- Case sensitivity
- Indentation
- Variables and data types
- Input/output
- If-else conditions
- Loops and functions
- Error handling

🎯 Homework for Practice

1. Write a program that asks for your name and prints a greeting.

2. Write a program to print numbers from 1 to 10 using a loop.
3. Create a dictionary with your name, age, and city, then print it.

If you found this lecture helpful, don’t forget to **like**, **comment**, and **subscribe** to *Jamia Pakistan*.
In the next lecture, we’ll cover **Data Types and Variables in detail**.

Allah Hafiz and Happy Coding!
