🐍 Python Programming – Complete Topics with Code & Examples

This document covers **Python from basic to advanced**, explained in **simple language** with **examples and programs**. It is suitable for **school students, beginners, and competitive exam preparation**.

1️⃣ Introduction to Python

What is Python?

Python is a **high-level, interpreted, object-oriented** programming language.

Features of Python

- Easy to learn & read
- Free and open source
- Platform independent
- Large library support

Python Uses

- Web Development
- Data Science
- AI & Machine Learning
- Automation
- Game Development

2️⃣ Installing Python & First Program

First Python Program

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

3️⃣ Variables & Data Types

Variables

```
x = 10
name = "Ram"
```

Data Types

- int → 10
- float → 10.5
- str → "Hello"
- bool → True / False

```
a = 10
b = 5.5
c = "Python"
d = True
print(a, b, c, d)
```

4️⃣ Input & Output

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

5️⃣ Operators

Types of Operators

- Arithmetic: + - * / %
- Relational: > < >= <= == !=
- Logical: and, or, not

```
a = 10
b = 3
print(a + b)
print(a > b)
```

6️⃣ Conditional Statements

if, elif, else

```
age = 18
if age >= 18:
print("Eligible to vote")
else:
print("Not eligible")
```

7️⃣ Loops

for Loop

```
for i in range(1, 6):
print(i)
```

while Loop

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

8️⃣ Control Statements

- break
- continue
- pass

```
for i in range(1, 10):
if i == 5:
break
```

```
print(i)
```

9️⃣ Functions

User Defined Function

```
def add(a, b):
return a + b
print(add(5, 3))
```

🔟 Python Data Structures

List

```
marks = [10, 20, 30]
print(marks)
```

Tuple

```
data = (1, 2, 3)
print(data)
```

Set

```
s = {1, 2, 3}
print(s)
```

Dictionary

```
student = {"name": "Ram", "age": 15}
print(student)
```

1️⃣1️⃣ String Handling

```
text = "Python"
print(text.upper())
print(text.lower())
```

```
print(text[::-1])
```

1️⃣2️⃣ File Handling

Write File

```
f = open("data.txt", "w")
f.write("Hello Python")
f.close()
```

Read File

```
f = open("data.txt", "r")
print(f.read())
f.close()
```

1️⃣3️⃣ Exception Handling

```
try:
a = 10 / 0
except ZeroDivisionError:
print("Cannot divide by zero")
```

1️⃣4️⃣ Object Oriented Programming (OOP)

Class & Object

```
class Student:
def __init__(self, name):
self.name = name
def show(self):
print("Name:", self.name)
```

```
s1 = Student("Ram")
s1.show()
```

OOP Concepts

- Encapsulation
- Inheritance
- Polymorphism
- Abstraction

1️⃣5️⃣ Modules & Packages

```
import math
print(math.sqrt(16))
```

1️⃣6️⃣ Python Libraries (Overview)

- NumPy
- Pandas
- Matplotlib
- Flask
- Django

1️⃣7️⃣ Useful Python Programs

Check Prime Number

```
n = 7
if n > 1:
for i in range(2, n):
if n % i == 0:
print("Not Prime")
```

```
break
else:
print("Prime")
```

Greatest of 3 Numbers

```
a, b, c = 10, 20, 5
print(max(a, b, c))
```

✅ Conclusion

Python is a **powerful and beginner-friendly language**. Learning Python opens doors to **IT jobs, automation, AI, data science, and web development**.