Python is a versatile, beginner-friendly programming language that's widely used in web development, data science, automation, AI/ML, and more. Here's a quick breakdown of what makes Python special and how you might start or deepen your learning:

◆ **Why Python?**

- **Easy syntax** – Reads almost like English
- **Huge ecosystem** – Libraries for almost anything (e.g., `numpy`, `pandas`, `flask`, `django`, `opencv`, `scikit-learn`)
- **Cross-platform** – Runs on Windows, macOS, Linux, etc.
- **Popular in AI/ML** – Used by companies like Google, Netflix, and NASA

Built-in Python Data Types

Python has several built-in data types categorized as:

1. Text Type

- **`str`**: String

2. Numeric Types

- **`int`**: Integer
- **`float`**: Floating point number
- **`complex`**: Complex numbers

3. Sequence Types

- **`list`**
- **`tuple`**
- **`range`**

4. Mapping Type

- **`dict`**

5. Set Types

- **`set`**

6. Boolean Type

- **`bool`**

Hands-On Examples

String (`str`)

```
python

name = "Python"
print(name)           # PYTHON
print(name[0])                # P
```

Integer (`int`)

```
python

x = 10
y = 3
print(x + y)      # 13
```

Float (`float`)

```
python

pi = 3.14159
radius = 2
area = pi * (radius ** 2)
print(area)  # 12.56636
```

List (`list`) – Mutable

```
python

fruits = ["apple", "banana", "cherry"]
fruits.append("orange")
print(fruits)      # ['apple', 'banana', 'cherry',
'orange']
del fruits[0]
print(fruits)      # ['banana','cherry','orange']
fruits[1]= 'mosambi'
print(fruits)      # ['banana','mosambi','orange']
```

Tuple (`tuple`) – Immutable

```
python

colors = ("red", "green", "blue")
print(colors[1])      # green

print(colors[0])      # red
# colors[1] = "yellow"  ✗ Error: Tuples are immutable
```

Range (`range`)

```
python

eg:1
      for i in range(1, 6):
            print(i, end=" ")   # 1 2 3 4 5
eg:2
      fruits = ['apple', 'banana', 'cherry']
      for i in range(len(fruits)):
            print(fruits[i])
```

Dictionary (`dict`)

```
Python

person = {"name": "Alice", "age": 25}
print(person["name"])      # Alice
person["age"] = 26
print(person)
```

Set (`set`)

```
python

numbers = {1, 2, 3, 3, 2}
numbers.add(4)
print(numbers)       # {1, 2, 3, 4}
```

Boolean (`bool`)

```
python

a = 5
b = 10
print(a < b)    # True
```

Conditional Statements

Conditional statements in Python let you execute code based on whether a condition is `True` or `False`. They're essential for decision-making in programs.

◆ Basic Conditional Statement (`if`, `elif`, `else`)

```
age = 18

if age < 18:
```

```
    print("You are a minor.")
elif age == 18:
    print("You are eligible for voting")
else:
    print("You are eligible for voting")
```

Loops in Python

1. For Loop

```
python

fruits = ["apple", "banana", "cherry"]

for i in fruits:
    print(i)
```

Print Numbers and Skip Multiples of 3

```
for i in range(1, 21):
    if i % 3 == 0:
        continue  # skip this iteration
    print(i)
```

Python code to create a dictionary of student names and their marks, and then print only those who scored more than 80:

```
python

# Creating a dictionary of students and their marks
students = {"Alice": 85,"Bob": 72,"Charlie": 90,
"David": 65,”Eva": 95}

# Printing students who scored more than 80
print("Students who scored more than 80:")
for name, marks in students.items():
    if marks > 80:
        print(f"{name}: {marks}")
```

Output:

```
Students who scored more than 80:
Alice: 85
Charlie: 90
Eva: 95
```

2. While Loop

```
python

count = 1

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

Find the Largest Number in a List

```
python

numbers = [4, 15, 2, 9, 22, 6]
largest = numbers[0]

for num in numbers:
    if num > largest:
        largest = num

print("The largest number is:", largest)
```

Functions

In Python, **functions** are reusable blocks of code that perform a specific task. They help make code modular, easier to read, and maintain.

 Types of Functions in Python**:**

1. Built-in Functions – Already available in Python
   Examples: `print()`, `len()`, `type()`, `range()`
2. User-defined Functions – Created by the programmer using the `def` keyword

 Syntax of a Function:

```
def function_name(parameters):
    # block of code
    return result  # optional
```

Example 1: Function with No Arguments and No Return Value

```
def greet():
    print("Hello, welcome to Python!")

greet()  # Call the function
```

Example 2: Function with Parameters

```
def add(a, b):
    print("Sum:", a + b)

add(5, 3)
```

Example 3: Function with Return Value

```
def multiply(x, y):
    return x * y

result = multiply(4, 5)
print("Product:", result)
```

Lambda Functions (Anonymous functions)

```
Eg 1:

square = lambda x: x * x
print(square(6))  # Output: 36

Eg 2:
def add(x, y):
```

```
    return x + y

# Lambda function
add_lambda = lambda x, y: x + y

print(add_lambda(3, 5))  # Output: 8
```

Examples:

1. Check if a Number is Even or Odd

```
def evenorodd(num):
    if num % 2 == 0:
        return "Even"
    else:
        return "Odd"

number = int(input("Enter a number: "))
print("The number is", evenorodd(number))
```

2. Find the Factorial of a Number

```
def factorial(n):
    result = 1
    for i in range(1, n + 1):
        result =result * i
    return result

num = int(input("Enter a number: "))
print("Factorial is", factorial(num))
```

---

3. Find the Largest of Three Numbers

```
def largest(a, b, c):
    return max(a, b, c)
```

```
x = int(input("Enter first number: "))
y = int(input("Enter second number: "))
z = int(input("Enter third number: "))
print("The largest number is:",largest(x, y, z))
```

4 . Count Character Frequency and save as in dictionary

```
python

text = "hello world"
char_freq = {}

for i in text:
    if i != " ":
        char_freq[i] = char_freq.get(i, 0) + 1

print("Character Frequency:", char_freq)
```

5.Pgm to check whether a number is prime or not

```
n = int(input("Enter a number: "))

if n > 1:
    for i in range(2, int(n/2) + 1):
        if n % i == 0:
            print(n, "is not a prime number")
            break
    else:
        print(num, "is a prime number")
else:
    print(num, "is not a prime number")
```