UNIT II

Introduction to Python

Python is a **high-level, interpreted, general-purpose programming language** created by **Guido van Rossum**. It is widely used in web development, data science, AI, automation, IoT, and more.

Applications of Python

- Web Development (Django, Flask)
- Data Science & Machine Learning
- Automation & Scripting
- IoT & Embedded Systems
- Game Development

Features of Python

- **Simple and Easy to Learn**
- **Interpreted Language**
- **Platform Independent**
- **Object-Oriented**
- **Large Standard Library**
- **Open Source**
- **Dynamic Typing**

Structure of a Python Program

A Python program consists of:

1. **Comments**
2. **Import Statements**
3. **Variable Declarations**
4. **Executable Statements**
5. **Functions**

Example:

```
# This is a Python program
import math

x = 10
print("Value of x:", x)
```

Python Syntax

- Python uses **simple English-like syntax**
- No semicolons (;) are required
- Code is executed line by line

Indentation

- Indentation defines **blocks of code**
- Mandatory in Python
- Usually **4 spaces** are used

Example:

```
if x > 0:
   print("Positive number")
```

Comments

Used to explain code.

- **Single-line comment:** #
- **Multi-line comment:** ''' comment ''' or """ comment """

Variables and Data Types

Variables

- Used to store data
- No need to declare type explicitly

```
a = 10
name = "Python"
```

Data Types

Type	Example
int	10
float	3.14
complex	2+3j
str	"Hello"
bool	True / False
list	[1,2,3]
tuple	(1,2,3)
set	{1,2,3}
dict	{"a":1}

Operators and Expressions

Types of Operators

- **Arithmetic:** + - * / % // **
- **Relational:** > < >= <= == !=
- **Logical:** and or not
- **Assignment:** = += -=
- **Bitwise:** & | ^ ~
- **Membership:** in, not in
- **Identity:** is, is not

9. Input and Output Functions

Input

```
name = input("Enter name: ")
age = int(input("Enter age: "))
```

Output

```
print("Name:", name)
```

10. Type Conversion

Converting one data type to another.

Types

- **Implicit Conversion**
- **Explicit Conversion**

Example

```
x = int("10")
y = float(5)
```

Decision Making

Used to execute code based on conditions.

Types

- if
- if-else
- if-elif-else

Example

```
if marks >= 50:
   print("Pass")
else:
   print("Fail")
```

Looping

Used to repeat a block of code.

Types of Loops

- **for loop**
- **while loop**

Example

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

Functions

A function is a reusable block of code.

Syntax

```
def add(a, b):
   return a + b
```

Lambda Expressions

Anonymous functions with one expression.

Syntax

lambda arguments : expression

Example

```
square = lambda x: x*x
print(square(5))
```

Scope of Variables

- **Local Scope:** Inside a function
- **Global Scope:** Outside all functions

Example

```
x = 10  # global

def func():
   x = 5  # local
   print(x)
```

Recursion

A function calling itself.

Example: Factorial

```
def fact(n):
   if n == 1:
      return 1
   return n * fact(n-1)
```

Introduction to Python IDEs

An IDE provides tools to write and debug code easily.

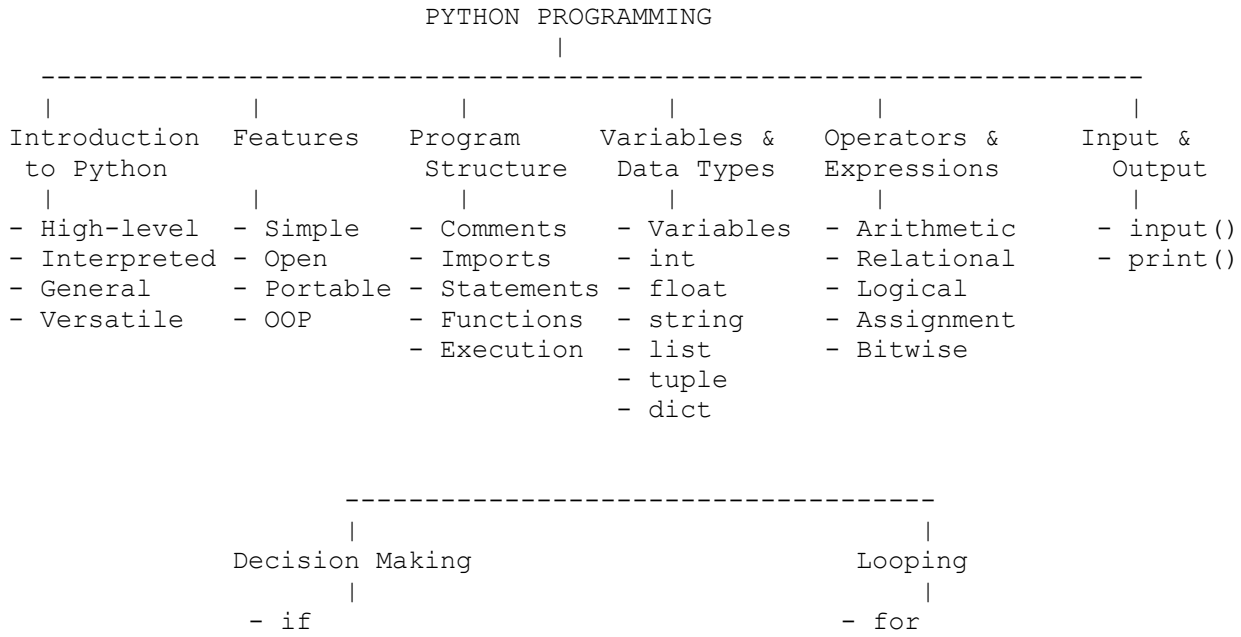
Popular Python IDEs

- **IDLE**
- **PyCharm**
- **VS Code**
- **Jupyter Notebook**
- **Spyder**

Advantages of Using Python IDEs

- Syntax highlighting
- Auto-completion
- Debugging tools
- Error detection

Mind Map: Python Programming Basics



```
              - if-else                           - while
              - if-elif-else                      - nested loops

         ----------------------------------------------------------------
         |                |                  |               |           |
      Functions         Lambda            Scope of        Recursion     IDEs
         |                |               Variables
   - def keyword      - Anonymous         - Local       - Function      -
IDLE
   - Parameters       - One expression    - Global        calling       -
PyCharm
   - return           - Short syntax                       itself       - VS
Code
                                                           - Base case -
Jupyter


               ------------------------------------
               |                                  |
           Syntax                           Type Conversion
               |                                  |
         - Simple rules                   - Implicit
         - No semicolons                  - Explicit (int, float, str)
         - Case-sensitive


                       -----------------
                       |               |
                  Indentation       Comments
                       |               |
                  - Mandatory       - Single-line (#)
                  - 4 spaces        - Multi-line (''' ''')
```