blem-solving

**Name- Varshaben Chaudhary**

**Project- Python Basic Training for Employees**

**Introduction to Python**

**What is Python?**

High-level, interpreted programming language

Known for readability and simplicity

Versatile for various applications (web development, data science, machine learning, automation)

**Why Python?**

Easy to learn and use

Large and active community

Extensive libraries and frameworks

Setting up Python Environment:

**Download and install Python** from https://www.python.org/downloads/

Verify installation by opening a command prompt and typing python --version

Consider using a code editor or IDE (Visual Studio Code, PyCharm, Jupyter Notebook)

**Basic Syntax Indentation**:

**Python uses** whitespace (indentation) to define code blocks

Consistent indentation is crucial for correct execution

Typically, 4 spaces are used

**Variables and Data Types:**

Variables store data values

No need to declare data types explicitly

**Common data types:**

Integers (whole numbers)

Floating-point numbers (decimal numbers)

Strings (text)

Booleans (True or False)

Example:

Python

name = "Alice"

age = 30

is\_student = False

**Operators:**

Arithmetic operators (+, -, \*, /, //, %, \*\*)

Comparison operators (==, !=, <, >, <=, >=)

Logical operators (and, or, not)

Example:

Python

result = 5 + 3

is\_greater = 10 > 5

**Control Flow**

**Conditional Statements**:

if, else, elif statements for decision-making

Indentation determines code blocks

Example:

Python

if age >= 18:

print("You are an adult")

else:

print("You are a minor")

**Loops:**

for loops for iterating over sequences

while loops for repeating code based on a condition

Example:

**Python**

**for i in range(5):**

print(i)

count = 0

while count < 3:

print("Count:", count)

count += 1

**Functions**

**Reusable blocks of code**

**Defined using the def keyword**

**Can take parameters and return values**

Example:

Python

def greet(name):

print("Hello,", name)

greet("Boby")

**Data Structures**

**Lists:**

Ordered collections of items

Mutable (can be changed)

Example:

Python

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

fruits.append("grape")

**Tuples:**

Ordered collections of items

Immutable (cannot be changed)

Example:

Python

coordinates = (3, 4)

**Dictionaries:**

Unordered collections of key-value pairs

Example:

Python

person = {"name": "Alice", "age": 30, "city": "New York"}

**Input and Output**

**Input:**

input() function to get user input

Example:

Python

name = input("Enter your name: ")

**Output:**

print() function to display output

Example:

Python

print("Hello,", name)

**Error and Exception Handling**

Use try and except blocks to handle errors

Common exceptions: ValueError, TypeError, IndexError

Example:

Python

try:

num = int(input("Enter a number: "))

except ValueError:

print("Invalid input")

Use code with caution.

**Additional Topics** (Optional)

Modules and packages

File handling

Object-oriented programming

Regular expressions

Libraries (NumPy, Pandas, Matplotlib)

Exercises and Practice

Provide hands-on exercises to reinforce learning

Encourage experimentation and pro