



1

# Chapter 1

## 🐍 Python Notes – Chapter 1: Basics + Practice

### 📄 1) `first.py` – First Program

#### 🧠 Concepts Covered:

- `print()` function
- Single-line and multi-line comments

#### 📖 Code:

```
print("HELLO WORLD") # Prints Hello World to the console  
# First program
```

```
""" This is a  
Multiple line comment """
```

```
# You can also comment multiple lines by selecting them and pressing Ctrl+/  
# Cool
```

```
# Awesome
# Fun
```



### Explanation:

- `print("HELLO WORLD")` : Basic output command in Python.
- `#` : Used for single-line comments.
- `""" """` or `''' '''` : Used for multi-line comments or docstrings.
- Shortcut for commenting multiple lines at once: `Ctrl + /` in most editors (like VS Code).



## 2) `Module.py` – Using External Modules



### Concepts Covered:

- Importing modules
- Using `pip` to install external libraries



### Code:

```
# Use PIP INSTALL (MODULE NAME) to install a module
import pyjokes

print("Sure, here is your joke")

# This prints a random joke
joke = pyjokes.get_joke()
print(joke)
```



### Explanation:

- `pip install pyjokes` : Used to install the `pyjokes` module (run in terminal).
- `import pyjokes` : Brings the module into your code.
- `pyjokes.get_joke()` : Fetches a random one-liner joke from the module.



**Real-world use:** Great for testing or making CLI apps fun!

### 3) Practice Problem 1 – Print a Poem

#### Concepts Covered:

- Multi-line strings
- Printing large text blocks

#### Code:

```
# We have used triple quotes ''' here to print multiple line code
```

```
print("""Twinkle Twinkle, Little Star  
How I wonder what you are  
Up above the world so high  
Like a diamond in the sky  
Twinkle Twinkle Little Star  
How I wonder what you are!  
Twinkle Twinkle, Little Star  
How I wonder what you are  
Up above the world so high  
Like a diamond in the sky  
Twinkle Twinkle Little Star  
How I wonder what you are""")
```

#### Explanation:

- Triple quotes `''' '''` allow you to write and print multi-line strings easily.
- Useful for poems, stories, or long-form text outputs.

#### Practice Question (Guessed):

Write a Python program to print a multi-line poem using a single `print()` statement.

### 4) Practice Problem 2 – Use Text-to-Speech Module

## Concepts Covered:

- Using external modules
- Text-to-speech synthesis

## Code:

```
# Use an external module and perform an operation

import pyttsx3

# Initialize the text-to-speech engine
engine = pyttsx3.init()

# Say a line of text using the engine
engine.say("Hi, I'm an AI Assistant developed by Prathamesh Nalge. "
           "He is my God. Please send your donation. Paytm par Ek crore rupiya p

# Process and speak the text
engine.runAndWait()
```

## Explanation:

- `pyttsx3` : Python library for text-to-speech conversion.
- `init()` : Starts the engine.
- `say()` : Prepares the text to be spoken.
- `runAndWait()` : Actually makes it speak.

## Practice Question (Guessed):

Use a Python module to create a basic AI assistant that speaks a sentence aloud.

## 5) Practice Problem 3 – List all Directory Files

## Concepts Covered:

- `os` module

- File system navigation



## Code:

```
import os # Built-in module for interacting with the operating system

# Specifying the directory of the content
directory_path = '/'

# Lists all the files in the specified directory
contents = os.listdir(directory_path)

# Print each file and directory name
for item in contents:
    print(item)
```








## Explanation:

- `os.listdir(path)` : Lists all files/folders in the path.
- `/` is root directory in UNIX-based systems. On Windows, try something like `C:\\` .
- `for item in contents` : Loops through the files and prints them.



## Summary: Key Concepts from Chapter 1

-  `print()` for output
-  Single-line (`#`) and multi-line (`''' '''`) comments
-  `pip` for installing external Python modules
-  Working with modules like `pyjokes` , `pyttsx3` , `os`
-  Writing practical and fun scripts (jokes, TTS, file listing)