📘 Absolute Path & Relative Path – Full 2-Page Notes

---

1. Introduction

In any computer system or project, files and folders are stored in a directory structure. To access these files, we use something called a **path**.
A *path* tells the computer the **location** of a file or a folder. Without paths, the system will not know where a particular file exists.

There are mainly two types of paths:

1. **Absolute Path**
2. **Relative Path**

Both look similar but work differently. Understanding them is important for programming, web development, file handling, and project structure.

---

2. Absolute Path

Definition

An **absolute path** is the **complete address** of a file or folder, starting from the **root directory** of the system.

It clearly shows where the file exists, no matter from which folder you are accessing it.

- In **Windows**, the root is `C:/`, `D:/` etc.
- In **Linux/Mac**, the root is `/`.

Absolute path is like giving the **entire location** of a house including city, street, and house number.

Key Features

- Always starts from **root directory**.
- Shows the **full and exact** file location.
- Does **not change**, even if your current working directory changes.
- Long but very clear.
- Used when a fixed path is required.

Examples

Windows

- C:/Users/Devi/Desktop/project/main.py
- D:/Photos/Trip/image1.jpg

Linux/Mac

- /home/devi/project/data/test.csv
- /var/www/html/index.html

Web URLs

- https://example.com/assets/logo.png

These paths always point to the exact same file.

Uses of Absolute Path

- System configuration files
- External file loading
- Full URLs in websites
- File references in servers
- When the file location will not change

---

3. Relative Path

Definition

A **relative path** is the address of a file **relative to the current working directory**.
It does **not** start from the root.
Instead, it shows the location based on where the user currently is in the project.

Relative path is like giving directions from where you currently are, instead of telling the full address.

Key Features

- Starts from **current folder**, not root.
- Short and easy to write.
- Path **changes** if the file or folder is moved.
- Mostly used in projects.
- Perfect for HTML, CSS, JS, Python, Django templates.

Examples

Assume current directory =

project/

Same folder file:

- index.html
- app.js

Subfolder file:

- images/logo.png
- css/style.css

Go to parent folder:

- ../data/info.txt

Go two levels up:

- ../../config/settings.py

Current folder:

- ./main.js

Uses of Relative Path

- Web development (HTML, CSS, JS)
- Images, CSS, JS linking in websites
- Django template file linking
- Files inside project structures
- When project is moved between systems

---

4. Difference Between Absolute & Relative Path

Feature	Absolute Path	Relative Path
Starts From	Root directory	Current folder
Length	Long	Short
Location Changes?	No	Yes
Clarity	Very clear	Depends on structure
Used For	System files, full URLs	Project files, internal linking
Example	C:/Users/Devi/app/home.html	../home.html

---

5. Real-Life Example

Absolute Path = Full Address

Like writing:
“House No. 24, Road No.5, Hyderabad, Telangana.”
This address never changes.

Relative Path = Directions From Current Place

Like telling:
“From the bus stop, go left and walk 100 meters.”
These directions only work if both people start from the same spot.

---

6. Summary

- **Absolute Path** gives a *full fixed address* from the root.
- **Relative Path** gives a *short address* from the current directory.
- Absolute paths do not change, relative paths do change.
- Both are important in programming and file management.
- Web development mostly uses **relative paths**, while system tasks use **absolute paths**.