**Memo: File Encryption & Decryption Scripts**

**encrypt.py – File/Folder Encryption**

This script encrypts every file inside a given directory (including subdirectories) using AES-256 in CBC mode with PKCS7 padding.

* **Key Derivation:** A key is derived from the user-supplied password using PBKDF2-HMAC-SHA256 with a random 16-byte salt and 100,000 iterations.
* **Encryption Flow:**
  1. Generate random salt (16B) + IV (16B).
  2. Pad file data to AES block size.
  3. Encrypt with AES (CBC mode).
  4. Overwrite the file with [salt + iv + encrypted data].
* **Usage:** Run the script, enter a password and directory path → all files inside are encrypted in place.

**decrypt.py – File/Folder Decryption**

This script reverses the encryption process, restoring files back to their original state if the correct password is provided.

* **Password Validation:** Before decrypting, the script tests the entered password against the first file. If it fails, it prints:
* [ERROR] Wrong password. Exiting without decrypting.

and stops immediately.

* **Decryption Flow:**
  1. Read salt (16B) + IV (16B) + ciphertext.
  2. Re-derive key from password and salt.
  3. Decrypt with AES (CBC mode).
  4. Remove PKCS7 padding.
  5. Overwrite file with original plaintext.
* **Usage:** Run the script, enter the same password and directory path → all files inside are decrypted in place.

**Notes**

* Works with **any file type** (.pdf, .docx, .jpg, .py, etc.) since operations are done in binary mode.
* Files are **overwritten in place**. Backup is recommended for testing.
* Password must match the one used during encryption; otherwise, decryption is blocked.