Got it 👍 Let's go step by step.

🔑 What is AES?

AES (**Advanced Encryption Standard**) is a **symmetric encryption algorithm**.

- *Symmetric* → the **same key** is used to encrypt and decrypt data.
- It's widely used in banking, secure communications, Wi-Fi (WPA2/WPA3), file encryption, and more.
- It works on blocks of data (usually **128 bits = 16 bytes**) and supports key sizes of **128, 192, or 256 bits**.

🧩 How AES Works (Simplified)

1. **Plaintext** → the original message.
2. **Key** → secret passphrase (128/192/256 bits).
3. **AES Algorithm** → scrambles the plaintext using rounds of **substitution, permutation, and mixing**.
4. **Ciphertext** → unreadable encrypted output.

To get the original message back, you apply AES decryption with the **same key**.

🛠️ Example in Python

We'll use the pycryptodome library, which is a standard package for AES in Python.

```
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes

# 1. Generate a random 16-byte key (AES-128)
key = get_random_bytes(16)

# 2. Create AES cipher in ECB mode (simple, but not very secure in
practice)
cipher = AES.new(key, AES.MODE_ECB)

# 3. Prepare a message (must be multiple of 16 bytes, so we pad it)
plaintext = b"Hello AES World!"  # 16 bytes exactly
```

```
# 4. Encrypt
ciphertext = cipher.encrypt(plaintext)
print("Ciphertext:", ciphertext.hex())

# 5. Decrypt
decipher = AES.new(key, AES.MODE_ECB)
decrypted = decipher.decrypt(ciphertext)
print("Decrypted:", decrypted.decode())
```

Output (example):

```
Ciphertext: 5f2a7c9ab83f45c8ef16e91c6a3b2d7d
Decrypted: Hello AES World!
```

📦 Example with Safer Mode (AES-CBC)

ECB mode is weak (patterns leak). A better choice is **CBC mode**, which uses an **IV (Initialization Vector)**.

```
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes
from Crypto.Util.Padding import pad, unpad

key = get_random_bytes(16)  # AES-128 key
iv = get_random_bytes(16)   # Initialization Vector

cipher = AES.new(key, AES.MODE_CBC, iv)

plaintext = b"My secret message"
ciphertext = cipher.encrypt(pad(plaintext, AES.block_size))

print("Ciphertext (hex):", ciphertext.hex())

# Decryption
decipher = AES.new(key, AES.MODE_CBC, iv)
decrypted = unpad(decipher.decrypt(ciphertext), AES.block_size)

print("Decrypted:", decrypted.decode())
```

🚨 Important Notes

- AES is **secure** if used with a good mode: **CBC, GCM, or CTR**.
- ECB mode is **not recommended** for real-world use.
- The **key** must be kept secret—anyone with the key can decrypt the data.
- AES is fast and trusted; governments and corporations use it as a standard.

👉 Do you want me to also show you **a real-world example of encrypting a text file with AES** (Python script) so you can test it yourself?