

## BASE64

```
C:\Users\ACER\Dropbox\PC\Documents\laboratories> base64.py ...
1  base64 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
2
3  text = input("Enter text: ")
4
5  print("8 Bit Binary:")
6  binary = ""
7
8  for ch in text:
9      num = ord(ch)
10     bits = format(num, '08b')
11     print(ch, "=", num, "=", bits)
12     binary += bits
13
14 print("\n6 Bit Binary:")
15
16 encrypted = ""
17 for i in range(0, len(binary), 6):
18     bits = binary[i:i+6]
19     if len(bits) < 6:
20         bits = bits + "0" * (6 - len(bits))
21     place = int(bits, 2)
22     letter = base64[place]
23     print(bits, "=", place, "=", letter)
24     encrypted += letter
25
26 print("\nEncrypted Text:", encrypted)
27
28 # --- DECRYPTION ---
29 print("\nDecrypted:")
30 binary2 = ""
31
32 for ch in encrypted:
33     place = base64.index(ch)
34     bits = format(place, '06b')
35     binary2 += bits
36     print(ch, "=", place, "=", bits) # show letter and binary
37
38 decrypted = ""
39 print("\nDecrypted 8-bit groups:")
40 for i in range(0, len(binary2), 8):
41     byte = binary2[i:i+8]
42     if len(byte) == 8:
43         letter = chr(int(byte, 2))
44         print(byte, "=", int(byte, 2), "=", letter)
45         decrypted += letter
46
47 print("\nDecrypted Text:", decrypted)
48
```

First, it defines the Base64 alphabet (a-z, 0-9, +, /). The user enters plaintext, then `format(ord(ch), '08b')` converts each character to 8-bit binary. These 8-bit segments are merged into a single lengthy binary string.

## ENCRYPTION

Next, the program divides the binary string into 6-bit groups since Base64 represents data in 6-bit chunks. If the last group has fewer than six bits, it inserts more zeros to complete it. Each 6-bit value is transformed to a decimal integer, which is then used as an index to locate the appropriate character in the Base64 alphabet. These characters make up the encrypted (encoded) Base64 text.

## DECRYPTION

To decrypt, the program reverses the procedure.

It translates each Base64 letter back to its 6-bit binary counterpart (depending on its index in the Base64 alphabet), then connects all of the 6-bit groups together to form 8-bit bytes. Each 8-bit chunk is then transformed back into an ASCII character using `chr(int(byte, 2))`, resulting in the reconstruction of the original plaintext.