Python チートシート

ファイルの内容を読み込む

ファイルの内容をbytearrayという可変長のバイト列に読みこんでいます。

```
# 例: encryptedというバイナリファイルをbytearrayに読み込む
file_path = "encrypted"  # 読み込むファイルのパス
with open(file_path , "rb") as f:
    data = bytearray(f.read())

# 読み込んだbytearrayを表示する
print(data)
```

2バイトずつ読み込んでいます。

```
file_path = "encrypted"
with open(file_path, "rb") as f:
    data = bytearray(f.read(2))
    while data:
        print(data)
        data = bytearray(f.read(2))
```

復号

加算暗号

加算暗号は、バイトデータに特定の値を加える暗号方式です。復号には、暗号化時に加算された値を減算します。

```
def decrypt_caesar (encrypted_data, key):
    return ''.join(chr((ord(char) - key) % 256) for char in encrypted_data)

encrypted_data = "ifmmp"  # 例: "hello" が key=1 で加算された結果
key = 1  # 暗号化時の加算値
decrypted_data = decrypt_caesar(encrypted_data, key)
print(decrypted_data)  # "hello"
```

XOR暗号

XOR暗号は、平文と鍵をビット単位でXOR演算する方式です。XOR演算の性質を使うと、同じ鍵で再度XOR演算することで元の平文を復号できます。

```
def xor_decrypt(encrypted_data, key):
    return ''.join(chr(c ^ key) for c in encrypted_data)

encrypted_data = [0x52, 0x5f, 0x56, 0x56, 0x55]  # XOR暗号化されたデータ（例）
key = 0x3A  # 1バイトの鍵
decrypted_data = xor_decrypt(encrypted_data, key)
print(decrypted_data) # "hello"
```

```
def xor_decrypt_with_key(encrypted_data, key):
    return ''.join(chr(encrypted_data[i] ^ key[i % len(key)]) for i in range(len(e

encrypted_data = [0x72, 0x5e, 0x76, 0x57, 0x75]  # 暗号化されたデータ
key = [0x1A, 0x3B]  # 複数バイトの鍵
decrypted_data = xor_decrypt_with_key(encrypted_data, key)
print(decrypted_data)
```

ROR1暗号

右ローテションシフト(ROR)は、ビットを右シフトし、最右ビットを最左に回す操作です。復号は、左にシフト(ROL)して元の状態に戻すことで行えます。

```
def rol(byte, count):
    return ((byte << count) & 0xFF) | (byte >> (8 - count))

def decrypt_ror1(encrypted_data):
    return ''.join(chr(rol(ord(char), 1)) for char in encrypted_data)

encrypted_data = "\x34\xb2\x36\x36\xb7"  # helloをROR1で暗号化したデータ
decrypted_data = decrypt_ror1(encrypted_data)
print(decrypted_data)
```