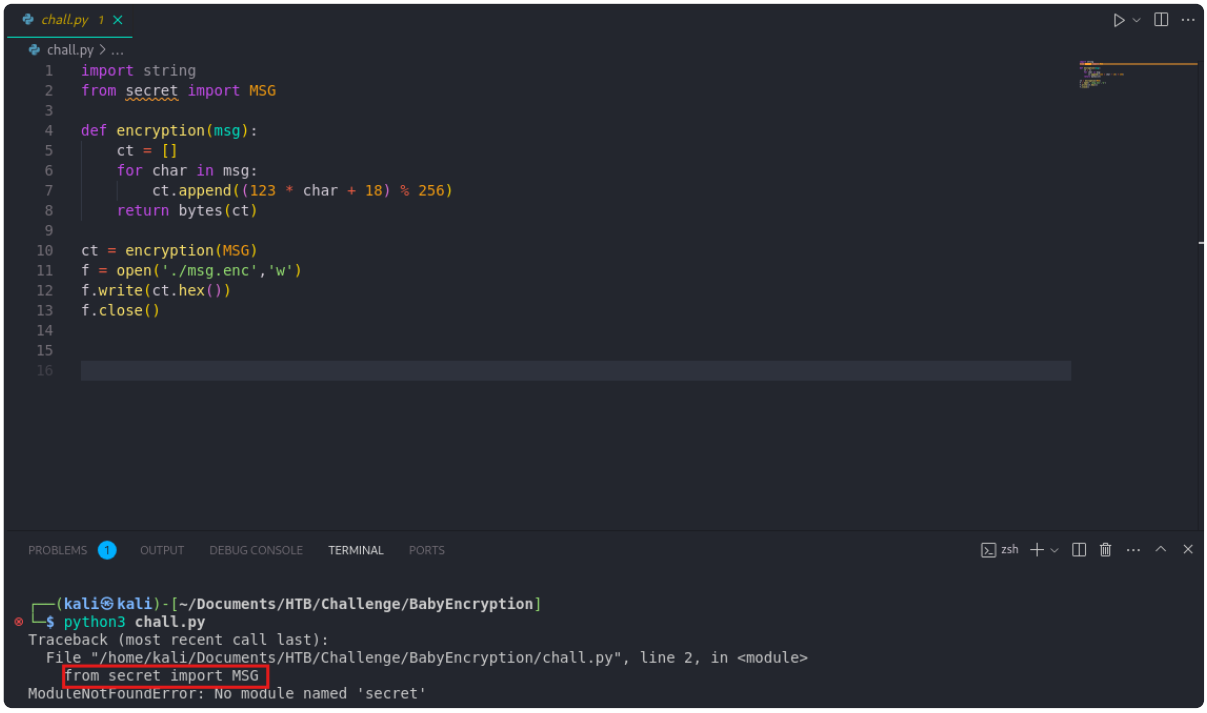
BabyEncryption

#crypto

1. Download the File
2. Execute the code and this is an error



3. Review source code
   - Each character of the MSG is multiplied by 123 and added to 18 in for loop.
   - The expression `(123 * val + 18) % 256` is performing a mathematical operation that likely plays a role in the encryption or decryption process.
   - Then modulus of 256 to make sure the character remains within the ASCII range.
   - The encrypted text is converted to Hexadecimal by hex function and stored in a File
   - `w` - write mode
4. To decrypt we need to reverse the Encryption process by brute-forcing the char values from 33-126 because the required characters of the flag in ASCII Range from 33-126

ct - also known for cypher text
Problem

```
import string
from secret import MSG

def encryption(msg):
    ct = []
    for char in msg:
        ct.append((123 * char + 18) % 256)
    return bytes(ct)

ct = encryption(MSG)
f = open('./msg.enc','w')
```

```
f.write(ct.hex())
f.close()
```

Solution

`bytes.fromhex(encrypt_text)` function converts a hexadecimal string into a bytes object.

```
encrypt_text =
"6e0a9372ec49a3f6930ed8723f9df6f6720ed8d89dc4937222ec7214d89d1e0e352ce0aa6ec82bf622227bb70e7fb7352249
b7d893c493d8539dec8fb7935d490e7f9d22ec89b7a322ec8fd80e7f8921"
result = ""

ct = bytes.fromhex(encrypt_text)

for char in ct:
    for val in range(33, 126):
        if ((123 * val + 18) % 256) == char:
            result += chr(val)
            break

print(result)
```

`chr()` - Convert inter to value which is between (33,126). Which is printable value

```
ascii_char = chr(97)
print (ascii_char) //Output 'a'
```