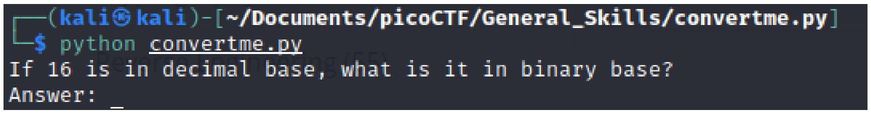
Write-up: Beginner picoMINI 2022 – convertme.py

This is another extremely beginner friendly CTF from picoCTF. They give us this description:

Run the Python script to convert the given number from the decimal to binary to get the flag.

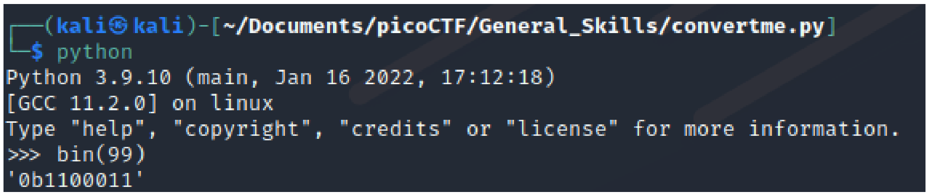
Capture

After creating the proper directories and downloading the file from picoCTF, I worked on getting the flag. Running the program results in this:



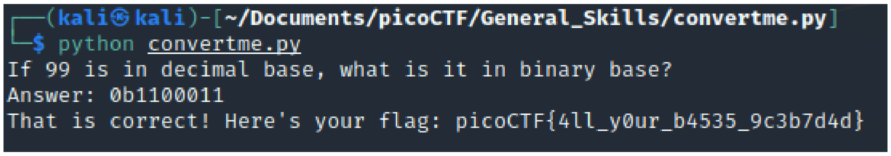
```
┌──(kali㉿kali)-[~/Documents/picoCTF/General_Skills/convertme.py]
└─$ python convertme.py
If 16 is in decimal base, what is it in binary base?
Answer: _
```

The value that convertme.py provided is 99. There are several ways to find the binary of this number. Division by two repeated until there are no remainders, the calculator app in Windows has a Programmer calculator that will work; I could also just look it up online. However, I do not think that is in the spirt of the task. I opt to use Python's built in function bin(). Feeding 16 into that function yields:



```
┌──(kali㉿kali)-[~/Documents/picoCTF/General_Skills/convertme.py]
└─$ python
Python 3.9.10 (main, Jan 16 2022, 17:12:18)
[GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> bin(99)
'0b1100011'
```

Feeding 0b1100011 into convertme.py yields:

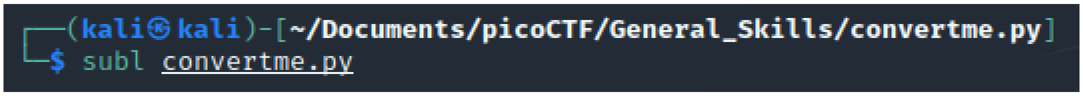


```
┌──(kali㉿kali)-[~/Documents/picoCTF/General_Skills/convertme.py]
└─$ python convertme.py
If 99 is in decimal base, what is it in binary base?
Answer: 0b1100011
That is correct! Here's your flag: picoCTF{4ll_y0ur_b4535_9c3b7d4d}
```

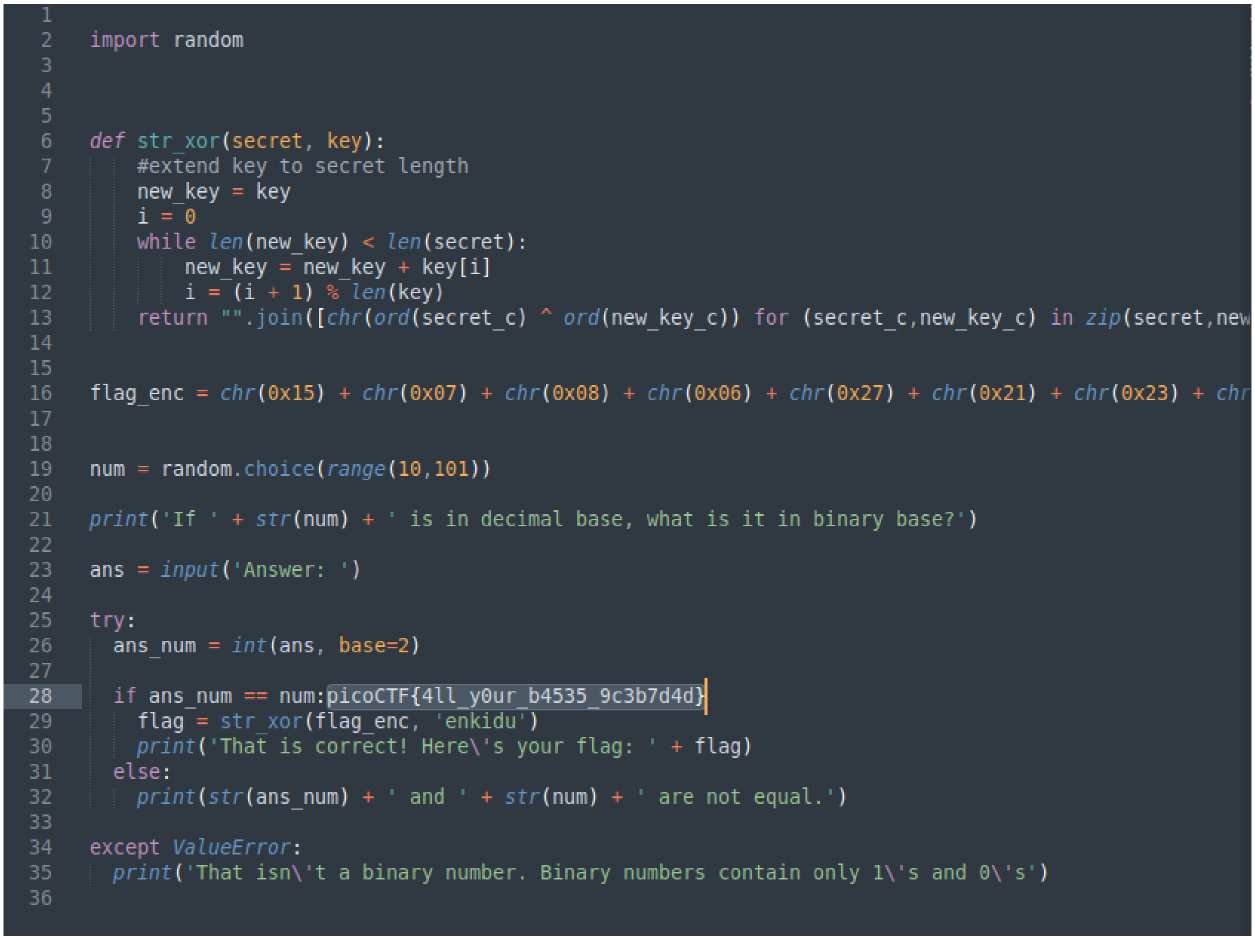
Mission complete: flag captured

Alternate Solve

In the spirit of hacking, I opened convertme.py in Sublime Text



```
┌──(kali㉿kali)-[~/Documents/picoCTF/General_Skills/convertme.py]
└─$ subl convertme.py
```



```python

import random



def str_xor(secret, key):
    #extend key to secret length
    new_key = key
    i = 0
    while len(new_key) < len(secret):
        new_key = new_key + key[i]
        i = (i + 1) % len(key)
    return "".join([chr(ord(secret_c) ^ ord(new_key_c)) for (secret_c,new_key_c) in zip(secret,new


flag_enc = chr(0x15) + chr(0x07) + chr(0x08) + chr(0x06) + chr(0x27) + chr(0x21) + chr(0x23) + chr


num = random.choice(range(10,101))

print('If ' + str(num) + ' is in decimal base, what is it in binary base?')

ans = input('Answer: ')

try:
  ans_num = int(ans, base=2)

  if ans_num == num:picoCTF{4ll_y0ur_b4535_9c3b7d4d}
    flag = str_xor(flag_enc, 'enkidu')
    print('That is correct! Here\'s your flag: ' + flag)
  else:
    print(str(ans_num) + ' and ' + str(num) + ' are not equal.')

except ValueError:
  print('That isn\'t a binary number. Binary numbers contain only 1\'s and 0\'s')

```

Bonus points?

Conclusion

This was another fun little challenge. I enjoyed being able to attack it from a different vector, as opposed to using Python’s built in functions. Thanks for reading.