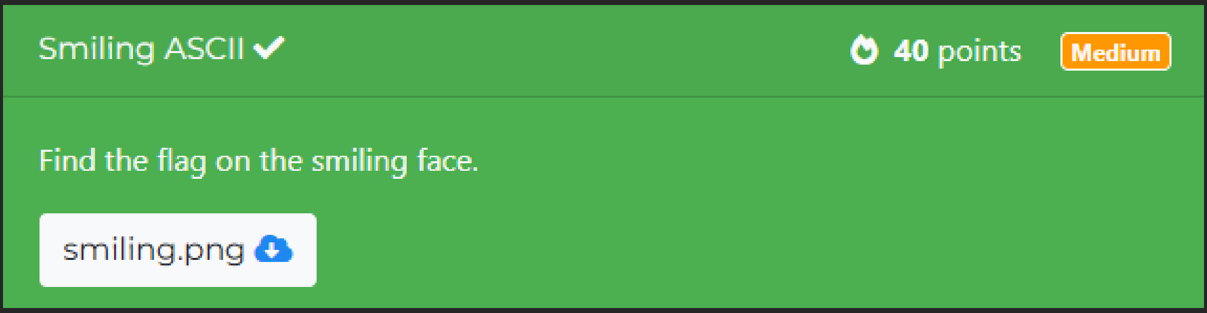
Smiling ASCII CTF Writeup

This document is a walkthrough on one way to solve the **Smiling ASCII CTF** on **CTFLearn**. The objective is to explain how I was able to solve this CTF to my future self.

General Information

- *Difficulty:* Medium
- *Category:* Forensics (Steganography)
- *Link:* Challenge - Smiling ASCII - CTFlearn - CTF Practice

Introduction

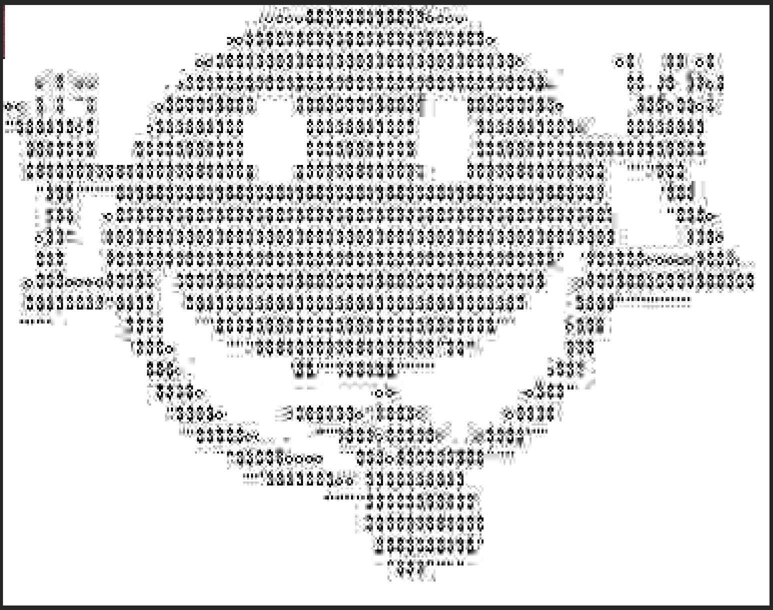


Smiling ASCII ✓ 40 points Medium

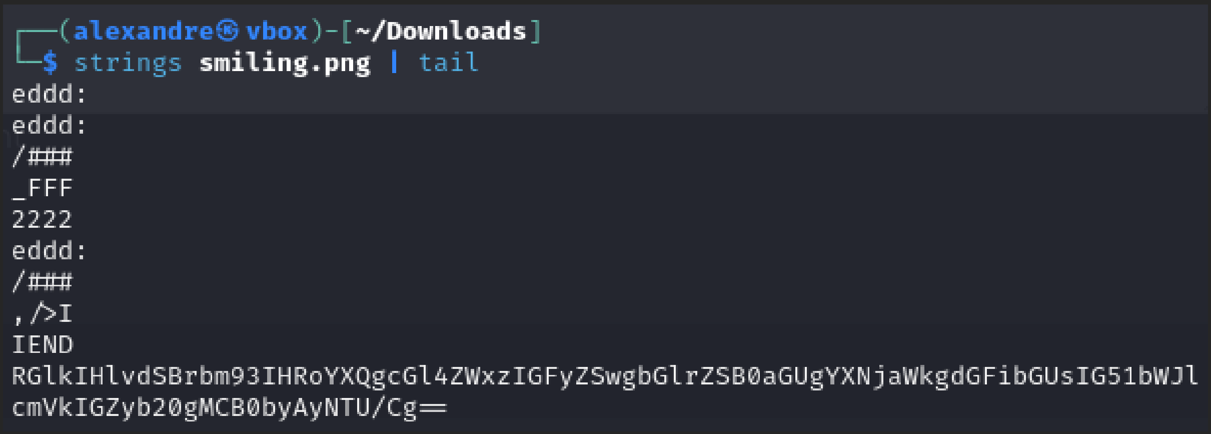
Find the flag on the smiling face.

smiling.png

We're given a PNG file, which looks like this:

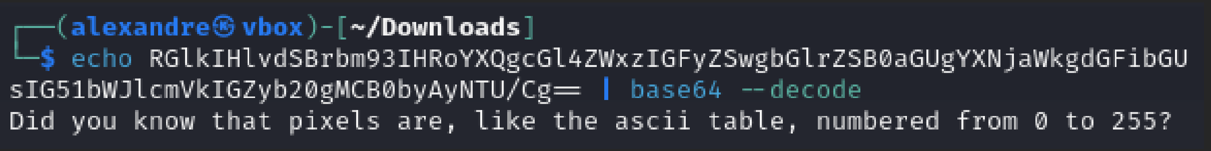


First and foremost, let’s do a simple strings command on the file to see if there’s anything hidden:



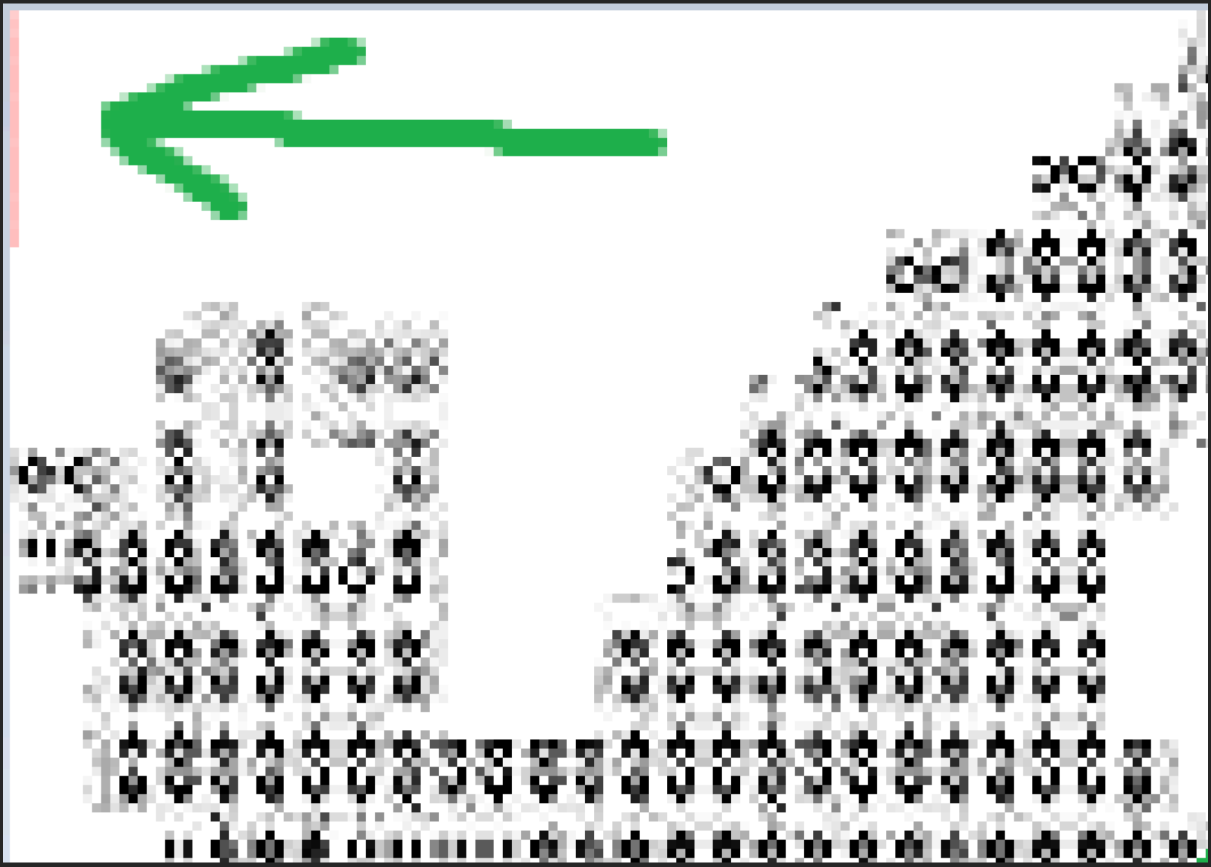
```
┌──(alexandre㉿vbox)-[~/Downloads]
└─$ strings smiling.png | tail
eddd:
eddd:
/###
_FFF
2222
eddd:
/###
,/>I
IEND
RGlkIHlvdSBrbm93IHRoYXQgcGl4ZWxzIGFyZSwgbGlrZSB0aGUgYXNjaWkgdGFibGUsIG51bWJl
cmVkIGZyb20gMCB0byAyNTU/Cg==
```

It looks like base64-encoded string was added at the end of the file

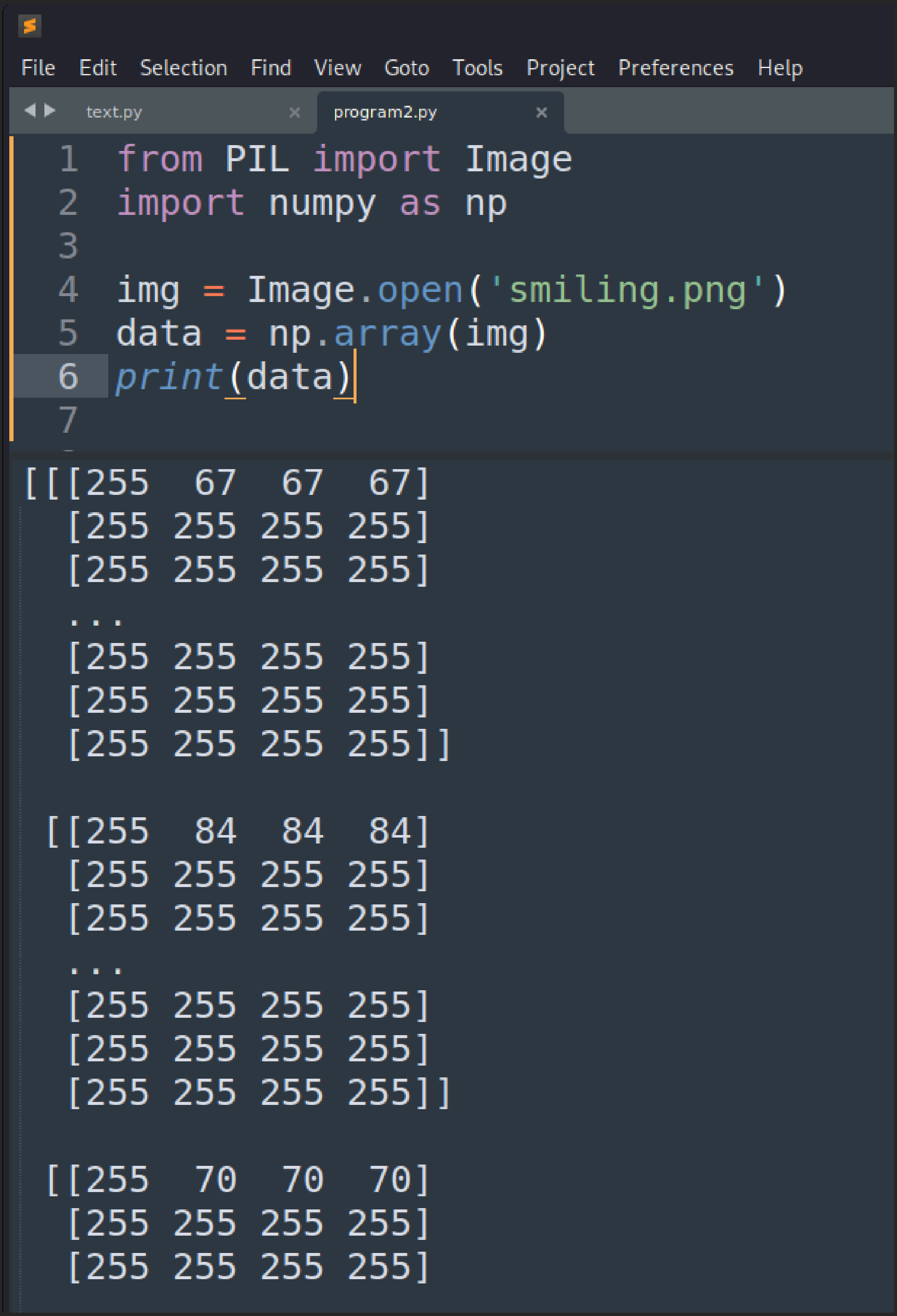


```
┌──(alexandre㉿vbox)-[~/Downloads]
└─$ echo RGlkIHlvdSBrbm93IHRoYXQgcGl4ZWxzIGFyZSwgbGlrZSB0aGUgYXNjaWkgdGFibGU
sIG51bWJlcmVkIGZyb20gMCB0byAyNTU/Cg== | base64 --decode
Did you know that pixels are, like the ascii table, numbered from 0 to 255?
```

It looks like we’re going to have to take a closer look at the pixels on the image
Effectively, upon closer look, red pixels can be seen at the top left of the image, which suggests that information has been added to the original image, perhaps the flag.



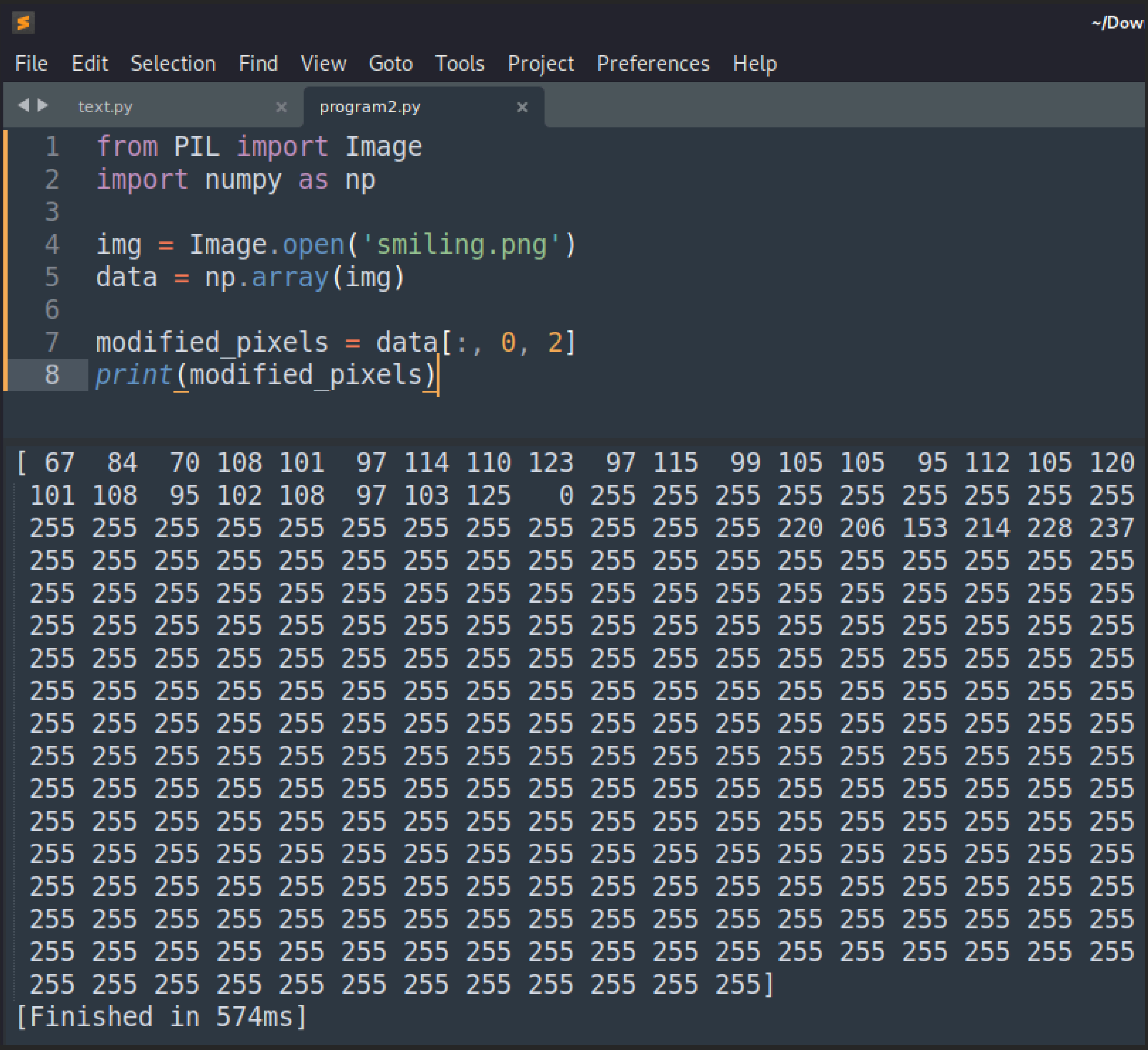
Using the PIL (for pillow) library in Python, we're able to convert the image into an array of numbers, each indicating the respective RGBA value of each pixel. The output confirms that the Green, Blue, and Alpha values of the first few pixels in the first row of the image are different to the others, perhaps their values are hiding a message ?



```python
from PIL import Image
import numpy as np

img = Image.open('smiling.png')
data = np.array(img)
print(data)
```

```
[[[255  67  67  67]
  [255 255 255 255]
  [255 255 255 255]
  ...
  [255 255 255 255]
  [255 255 255 255]
  [255 255 255 255]]

 [[255  84  84  84]
  [255 255 255 255]
  [255 255 255 255]
  ...
  [255 255 255 255]
  [255 255 255 255]
  [255 255 255 255]]

 [[255  70  70  70]
  [255 255 255 255]
  [255 255 255 255]
```

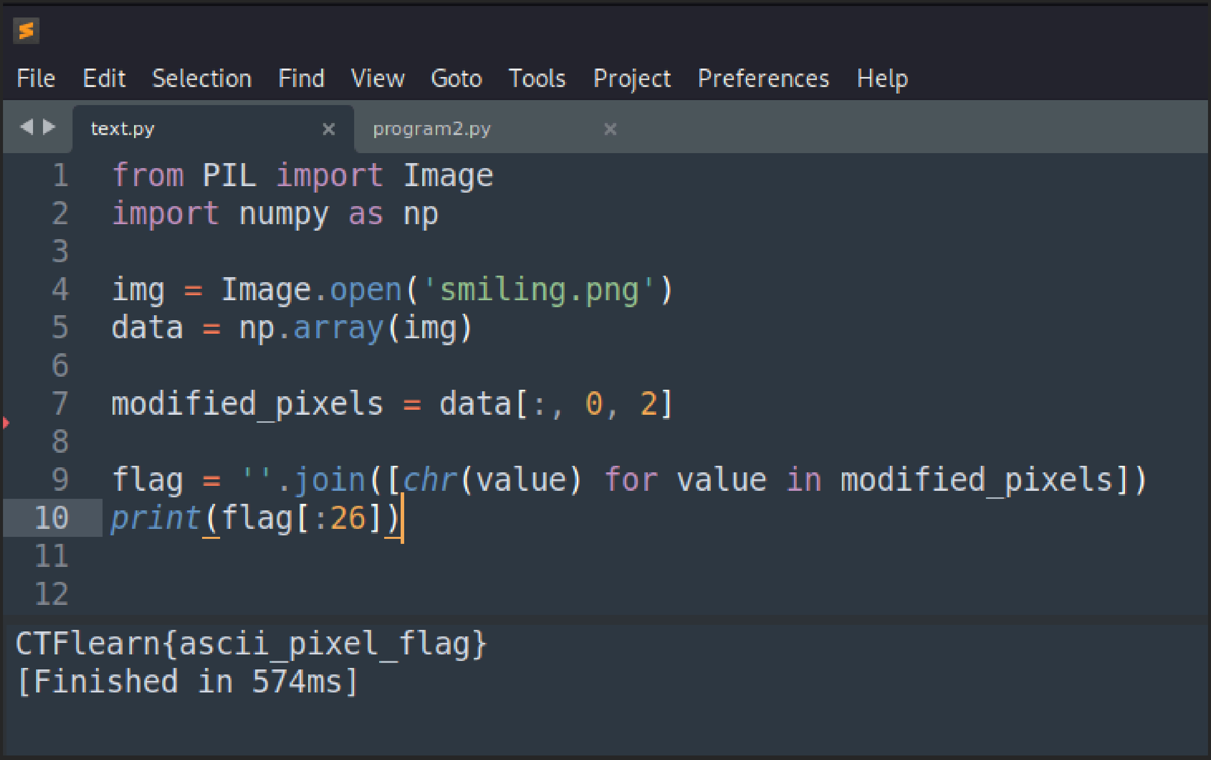
Let’s extract the blue value of each pixel in the row of the image (the same can be done for green and alpha values), we get the following array:


~/Dow
File Edit Selection Find View Goto Tools Project Preferences Help
text.py
program2.py

```python
from PIL import Image
import numpy as np

img = Image.open('smiling.png')
data = np.array(img)

modified_pixels = data[:, 0, 2]
print(modified_pixels)
```

```
[ 67  84  70 108 101  97 114 110 123  97 115  99 105 105  95 112 105 120
 101 108  95 102 108  97 103 125   0 255 255 255 255 255 255 255 255 255
 255 255 255 255 255 255 255 255 255 255 255 255 220 206 153 214 228 237
 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
 255 255 255 255 255 255 255 255 255 255 255 255]
[Finished in 574ms]
```

And there you have it, by converting the first 26 values of the array into ASCII characters, we find the following flag: **CTFlearn{ascii_pixel_flag}**.


File Edit Selection Find View Goto Tools Project Preferences Help
text.py
program2.py

```python
from PIL import Image
import numpy as np

img = Image.open('smiling.png')
data = np.array(img)

modified_pixels = data[:, 0, 2]

flag = ''.join([chr(value) for value in modified_pixels])
print(flag[:26])
```

```
CTFlearn{ascii_pixel_flag}
[Finished in 574ms]
```