0:

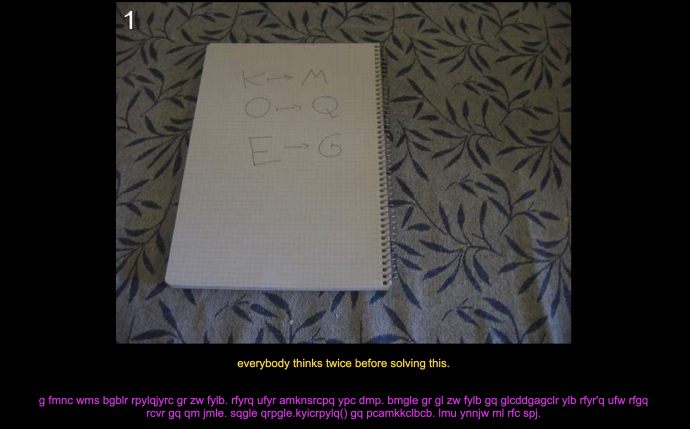


I can see that level 0 is displayed in the upper left corner, and the following prompt try to change the URL address, let us change the URL address, because the following levels will enter the next level in this way. 2^38 is displayed in the middle of the screen. It is guessed that the calculation result needs to be used to change the URL, so the code:

2\*\*38

The result is 274877906944, replace the 0 in the original URL to get the new URL.

1:



There are three sets of corresponding letters in the picture, and at the same time you can see a line of unreadable letters under the picture.

The current URL is http://www.pythonchallenge.com/pc/def/map.html

At this time, I guess it is to replace the map. The replacement rule is to push each letter back by two.

I use programming to translate the following text:

s='''g fmnc wms bgblr rpylqjyrc gr zw fylb. rfyrq ufyr amknsrcpq ypc dmp. bmgle gr gl zw fylb gq glcddgagclr ylb rfyr'q ufw rfgq rcvr gq qm jmle. sqgle qrpgle.kyicrpylq() gq pcamkkclbcb. lmu ynnjw ml rfc spj.'''

for i in range(len(s)):

if ord(s[i])>=ord('a') and ord(s[i])<=ord('x'):

c=chr(ord(s[i])+2)

print(c,end='')

elif ord(s[i])>=ord('y') and ord(s[i])<=ord('z'):

c=chr(ord(s[i])-24)

print(c,end='')

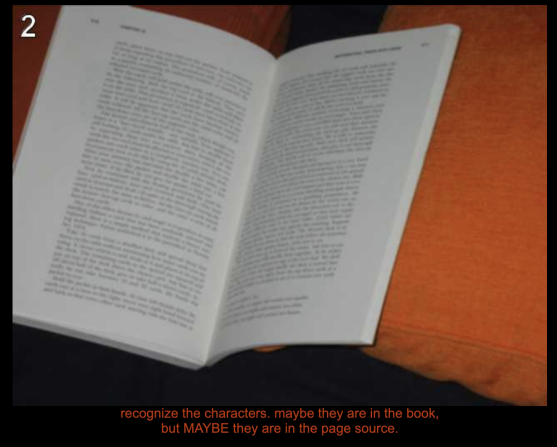
else:

print(s[i],end='')

The output result is:

i hope you didnt translate it by hand. thats what computers are for. doing it in by hand is inefficient and that's why this text is so long. using string.maketrans() is recommended. now apply on the url.

It prompts us to use string.maketrans( ) to perform calculations, but in fact we are almost sure of the answer. Obviously, it is enough to replace map with ocr, so the result is:



2:

I can see the prompt to view the source code of the webpage, right click to view the source code, I found a comment <!-- find rare characters in the mess below -->

Below is a very long character, about ten to twenty thousand, like this:



I program these rare characters:

from collections import Counter

Strings

c = Counter(strings)

print(c.most\_common( ))

I found that some letters appeared once, and the results are as follows:

[(')', 6186), ('@', 6157), ('(', 6154), (']', 6152), ('#', 6115), ('\_', 6112), ('[', 6108), ('}', 6105), ('%', 6104), ('!', 6079), ('+', 6066), ('$', 6046), ('{', 6046), ('&', 6043), ('\*', 6034), ('^', 6030), ('\n', 1219), ('e', 1), ('q', 1), ('u', 1), ('a', 1), ('l', 1), ('i', 1), ('t', 1), ('y', 1)]

from collections import Counter

strings

c = Counter(strings)

print(c.most\_common())

print(''.join([i[0] for i in c.items() if i[1]==1]))

The result is: equality

So get the new URL:



3:

I can see the prompt One small letter, surrounded by EXACTLY three big bodyguards on each of its sides.

The requirement is to find lowercase letters, which must have 3 uppercase letters on both sides.

I right-clicked to view the source code of the webpage and saw a long and complicated comment.

At this time, I noticed that the title of the page is re, so I use regular expressions to find:

（Because there can only be three uppercase letters on both sides, we add [a-z] on both sides.）

import re

strings

reg=re.compile('[a-z][A-Z]{3}[a-z][A-Z]{3}[a-z]')

print(''.join(reg.findall(dat)))

The result is this:

qIQNlQSLi eOEKiVEYj aZADnMCZq bZUTkLYNg uCNDeHSBj kOIXdKBFh dXJVlGZVm gZAGiLQZx vCJAsACFl qKWGtIDCj

The lowercase letter in the middle is what I'm looking for, modify the code:

result=reg.findall(dat)

url=''

for res in result:

url+=res[4]

print(url)

The result is: linkedlist, get the new URL.

After I opened it, I found that there was only one sentence linkedlist.php on the page, and I chose to replace the URL again:



4:

<html>

<head>

<title>follow the chain</title>

<link rel="stylesheet" type="text/css" href="../style.css">

</head>

<body>

<!-- urllib may help. DON'T TRY ALL NOTHINGS, since it will never

end. 400 times is more than enough. -->

<center>

<a href="linkedlist.php?nothing=12345"><img src="chainsaw.jpg" border="0"/></a>

<br><br><font color="gold"></center>

Solutions to previous levels: <a href="http://wiki.pythonchallenge.com/"/>Python Challenge wiki</a>.

<br><br>

IRC: irc.freenode.net #pythonchallenge

</body>

</html>

I saw the following two lines of text, provided a library urllib, and then told me not to try everything because it is infinite, and 400 cycles are enough.

<!-- urllib may help. DON'T TRY ALL NOTHINGS, since it will never end. 400 times is more than enough. -->

The text displayed after clicking the picture is "and the next nothing is 44827".

This clearly shows the usefulness of webscraper, prompting me to use the urllib library, but I still use requests to solve the problem.

import requests

url='http://www.pythonchallenge.com/pc/def/linkedlist.php?nothing=12345'

for i in range(400):

url='http://www.pythonchallenge.com/pc/def/linkedlist.php?nothing='+txt.split()[-1]

r=requests.get(url)

txt=r.text

print(txt)

I started printing the result when the loop was 400th time, but it didn't work. I tried several times before and after 400 and it didn't work. So I printed it out. It was printed out once about 250. So the result was like this:

and the next nothing is 41643

and the next nothing is 23416

and the next nothing is 54432

and the next nothing is 4448

... Many lines are omitted here

peak.html

and the next nothing is 72758

and the next nothing is 71301

So i got the new URL:



5:

Pronounce it, view the source code:

<html>

<head>

<title>peak hell</title>

<link rel="stylesheet" type="text/css" href="../style.css">

</head>

<body>

<center>

<img src="peakhell.jpg"/>

<br><font color="#c0c0ff">

pronounce it

<br>

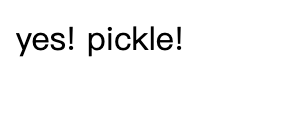
<peakhell src="banner.p"/>

</body>

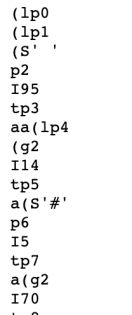
</html>

<!-- peak hell sounds familiar ? -->

I can see the comment <!-- peak hell sounds familiar? --> This is the pickle library, get the URL:



The title of the webpage is peak hell, and the source code of the webpage also has a peakhell tag, as follows:



So the code is:

import pickle

from urllib.request import urlretrieve

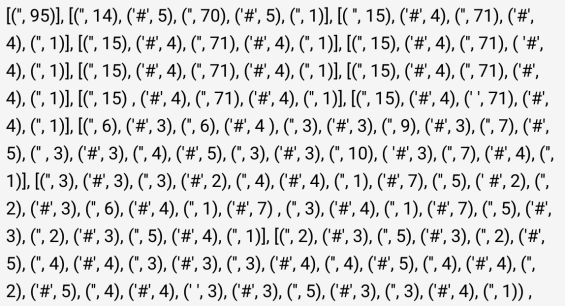
url = 'http://www.pythonchallenge.com/pc/def/banner.p'

urlretrieve(url, filename = 'five.pkl')

with open('five.pkl', 'rb') as f:

r = pickle.load(f)

The result is probably this:



I analyze the number should represent the number of characters, change the code:

from urllib.request import urlretrieve

import pickle

url = 'http://www.pythonchallenge.com/pc/def/banner.p'

urlretrieve(url, filename = 'five.pkl')

with open('five.pkl', 'rb') as f:

r = pickle.load(f)

for i in r:

for j in i:

print(j[0] \* j[1], end = '')

print('\n')

The printed result is the word channel.

So I got the result:



6:

“Donate” appears on the page. I check the source code and have comments as follows:

<!-- The following has nothing to do with the riddle itself. I just

thought it would be the right point to offer you to donate to the

Python Challenge project. Any amount will be greatly appreciated.

-thesamet

-->

There is also a comment in the source code <!-- <-- zip --> The zipper of the picture is also zip, it should be in the right direction. In fact, his other meaning is to prompt the compressed package.

After I downloaded the compressed package and opened it, there are more than 900 txt files, the last one is read me.

welcome to my zipped list.

hint1: start from 90052

hint2: answer is inside the zip

import zipfile as zi

The code is:

path = "C:\\Users\\sky\\Desktop\\channel.zip"

# Use the zipfile package to decompress and read the contents of the file to files

files = {}

fzip = zi.ZipFile(path)

for name in fzip.namelist():

with fzip.open(name) as fz:

files[name] = fz.read().decode("utf-8")

# The initial value of nothing in readme.txt

nothing = "90052"

while True:

f = nothing + ".txt"

strs=str(files[f])

print(strs)

try:

nothing = strs.split()[-1]

except:

break

The output is like this:

（Omit n lines here）

Next nothing is 67824

Next nothing is 46145

Collect the comments.

Prompt to collect comments, I go to find information. In fact, although the above code can output results, there are certain problems when solving the following problems, so I am correcting it.

import re

The code above

while True:

fz = nothing + ".txt"

# Get comment and output

print(fzip.getinfo(fz).comment.decode("utf-8"), end="")

if fz in files:

# print(files[fz])

result = re.search(r"Next nothing is (\d+)", files[fz])

try:

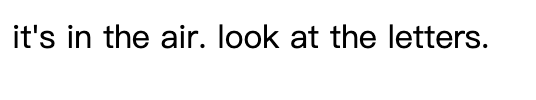
nothing = result.group(1)

except:

break

The result shows the word hockey.

Prompt after opening the link:



The above result is composed of O X Y G E N. Obviously, this is oxygen, so the result is:



7:

I use PIL to process the image, the mosaic is in the middle, and the programming is as follows:

from PIL import Image

im = Image.open("oxygen.png") #file path

width, height = im.size # Get size

pic = im.load()

h = height // 2

for x in range(width):

print(pic[x,h])

Result:

Omit...

(97, 97, 97, 255)

Omit...

(100, 100, 100, 255)

Omit...

(116, 116, 116, 255)

Omit...

I can see that R, G, and B are the same, maybe ASCII code.

from PIL import Image

im = Image.open("oxygen.png")

width, height = im.size

pic = im.load()

h = height // 2

ss=0

for x in range(width):

r, g, b, x = pic[x, h]

if r != g:

continue

if ss!=r:

print(chr(r), end='')

ss=r

The result is:

smart guy, you made it. the next level is [105, 10, 16, 101, 103, 14, 105, 16, 121]

answer = [105, 110, 116, 101, 103, 114, 105, 116, 121]

for l in answer:

print(chr(l), end='')

Output integrity. Get the URL:



Python Challenge is a very interesting web breakthrough game. Players can solve problems and upgrade their level through Python programming. I think this game is a test of a comprehensive grasp of Python. For example, some levels require regular expressions, and some levels use simple webscraper. This knowledge relates to week 2.

Level 0 is about the knowledge of Python's basic numerical operations. Open an IDE, such as Python's own terminal, and you can calculate the result with one line of code.

Level 1 examines the knowledge of string encoding and for loops.

The second level is to find the fewest characters in the following large string of characters. Regular expressions extract characters, list counts, and conditional statements.

Level 3 uses Requests to request a web page and then uses a regular expression to extract the string, and then the for loop counts the number of occurrences of each character.

At the beginning of level 4, you need to use python skills to solve it. This level starts to add defects. You need to manually enter the value into the URL and press Enter. It will repeatedly insert new values for you to enter and replace them in cycles. Here we need to use Python to realize auto-fill, modify the URL and replace it with the new URL, and loop until the webpage can no longer be replaced. I use webscraper and regular expressions to solve it. Extract the value from each webpage and replace it with a new URL and then request the webpage, loop down, you can get the answer.

At this point, I feel more and more that Python is a very practical scripting language. Although learning Python is difficult, it requires a lot of practice to put what I have learned into practice.