

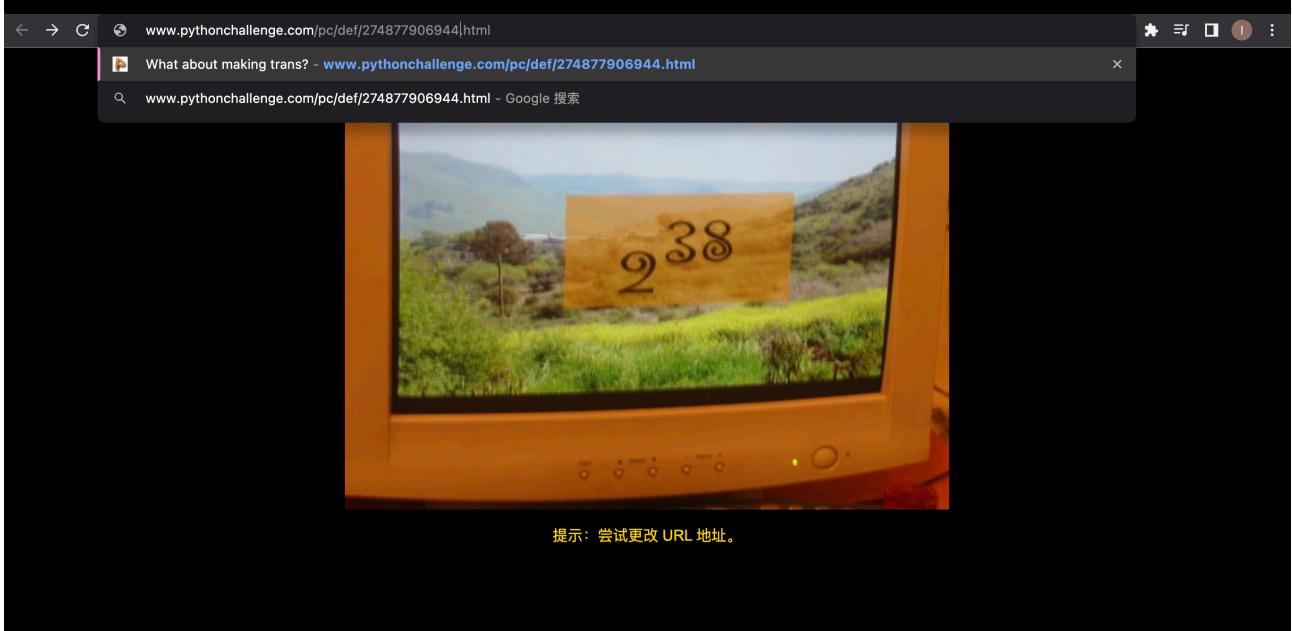
pythonchallenge

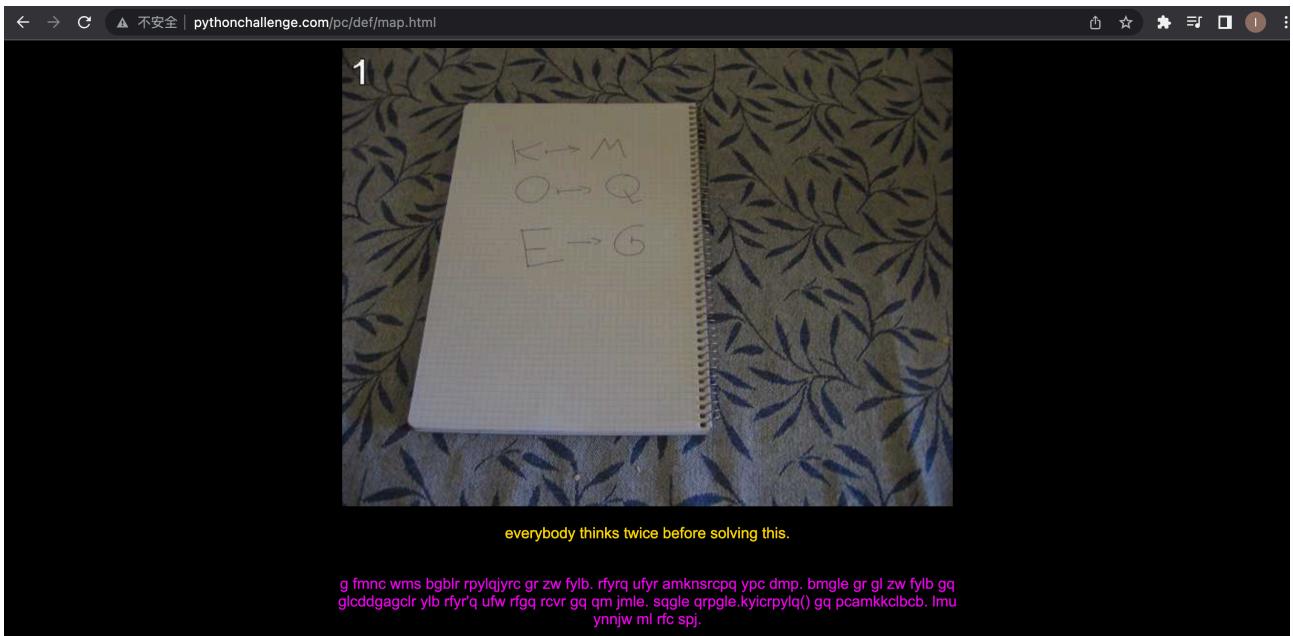
Level 0

How to pass the customs: The first time I saw this figure, I was a little at a loss, then I saw the Hint below that it was to change the URL address. At this time, I noticed that the level I was challenging was level 0, and the URL address above was 0.html. I thought it was related to numbers, because the middle one looked like 238, and it looked like 2 to the power of 38. So I tried to replace the 0 with 238 and 2 to the 38.

Solution: calculate 2 what is the power of 38, and then replace the URL in the 0, make web site <http://www.pythonchallenge.com/pc/def/274877906944.html>

Pass through pictures:



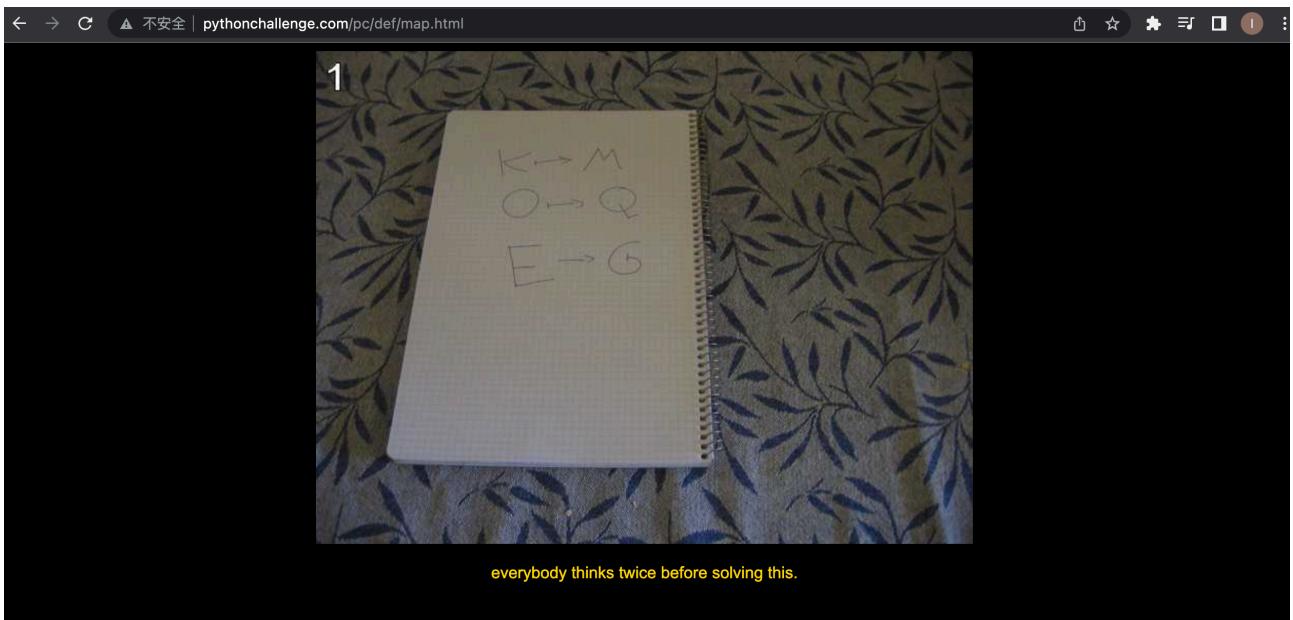


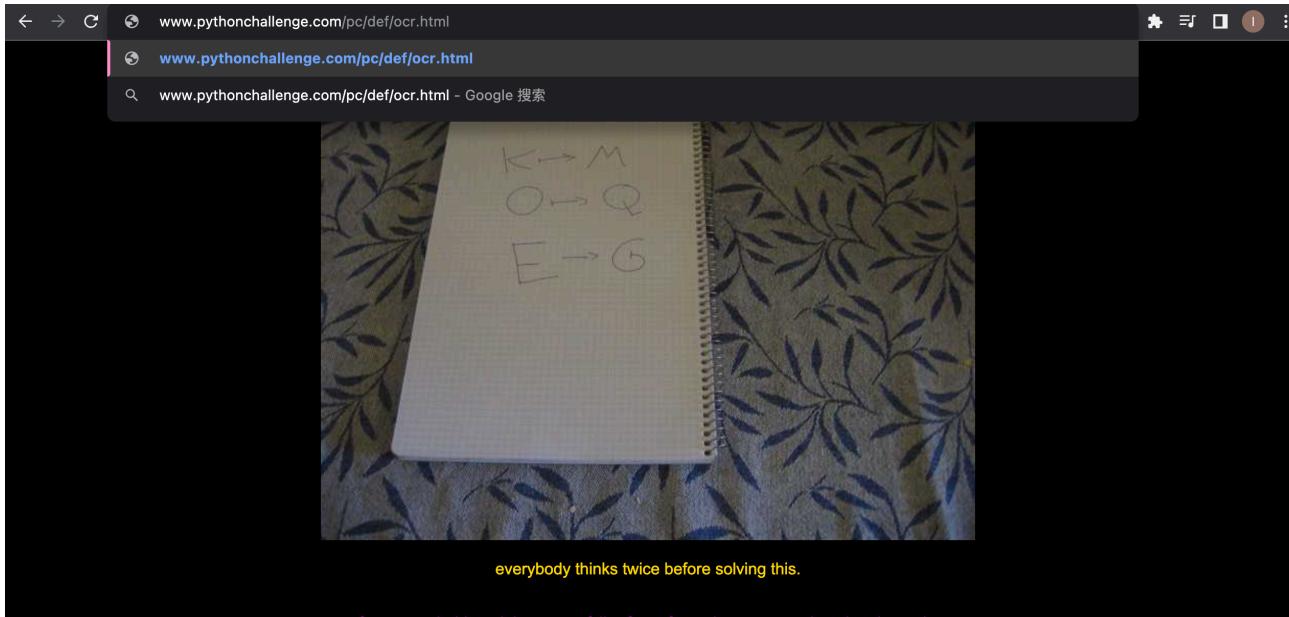
Level 1

How to pass the customs: The picture above shows K->M O->Q E->G, below a string of letters. I look not to come out in the beginning the following a string of letters is what meaning, but according to a close, whether I guess 'map' of the <http://www.pythonchallenge.com/pc/def/map.html> to convert into something else. I guess if I should move the letter two places back based on 'K->M O->Q E->G'. In accordance with this idea I will be the following string of letters to be translated out, and verified my guess.

Solution: Follow K->M ->Q -, E->G and move each letter two letters back to "ocr".

Pass through pictures:





Level 2

How to pass the customs: "Know the character" is displayed on the page. Maybe they're in the book, but maybe they're in the page source code." . Obviously, you can't read anything in this picture book, so you have to start with the page source code. When you enter the page source code, you find a comment below <! -- find rare characters in the mess below: -- > Guess that you might need to find rare characters in the long list of code below:

Solution: Follow the prompts to find rare characters in python. Found that get is "equality" and get new url:<http://www.pythonchallenge.com/pc/def/equality.html>

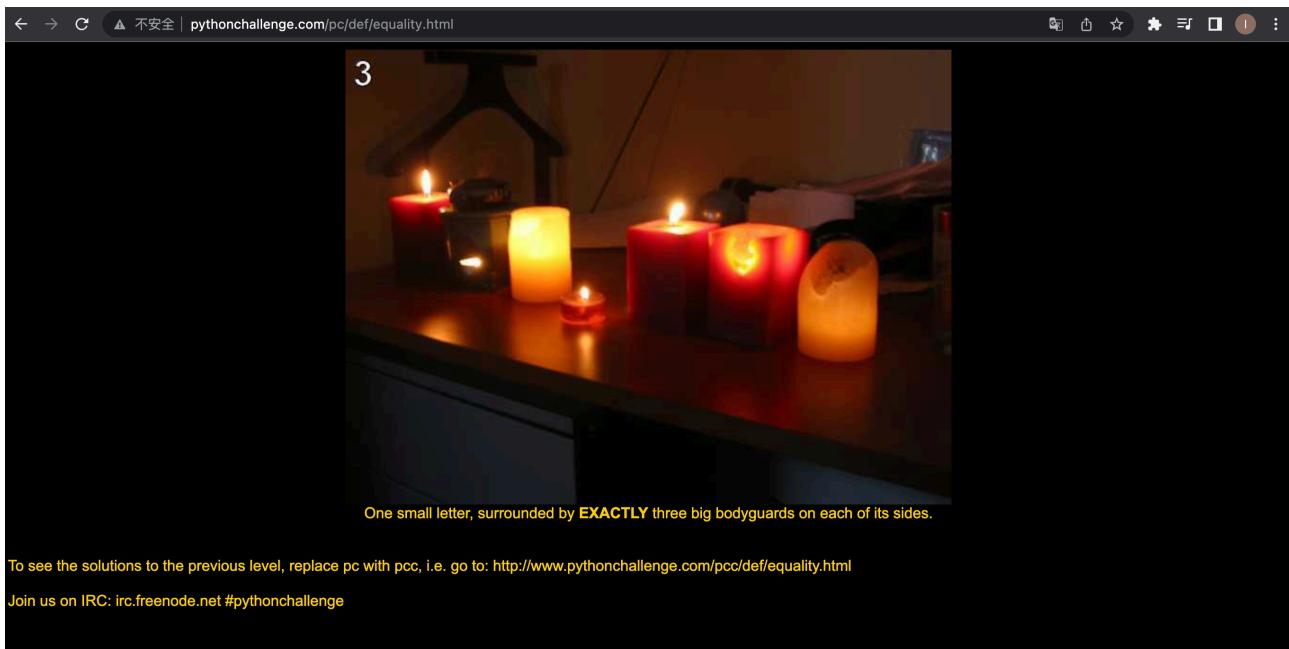
Pass through pictures:

Level 3

How to pass the customs: See the following prompt: "One small letter, surrounded by EXACTLY three big bodyguards on each of its sides." It's easy to understand, but it doesn't say where to look. Based on the experience of the previous level, I guess it will be from the web source code.

Solution: Open the page source code, displayed a string of characters. Use python to quickly find the letters that match the criteria.

Pass through pictures:



```
自动执行 □
1 <head>
2   <title>re</title>
3   <link rel="stylesheet" type="text/css" href="..../style.css">
4 </head>
5 <body>
6   <center>
7     <font color="#ff0000">
8       One small letter, surrounded by <b>EXACTLY</b> three big bodyguards on
9       each of its sides.
10    <br><br>
11    To see the solutions to the previous level, replace pc with pcc, i.e. go
12    to: http://www.pythonchallenge.com/pcc/def/equality.html
13
14    <br><br>
15    Join us on IRC: irc.freenode.net #pythonchallenge
16  </body>
17 </html>
18
19 k4ewtloYgcFqaJNhHVGxXDiQmzjfcpYbzxlWrVcqsmUbCunkfxzWdzjUZMiGghRRIUwGmYmvnJ1HEmbT
20 MUKLECKdCtheSYBpIElRnZugFAXDRtQPpceBgbfaRVvvguRLvxkAdLoeCKxsDuvBCwdpMMwuElEg
21 ENihrpCLhujoBgPDRPvzcrwadMMbkmkzCCzotPfDrizBqMblmxrXhNlCufprWXxHSpplkoLcrJd
22 vuyFCZtqKLHwIyzoXeglkxhVJImeuySguVmTCyMshQtvzpw1bd0HnbauwvUYCmgznObgByPw
23 TDQheAbsaMljTmaOKmNsLziVmEnfXqdATQ1jtwtcyHeMwQTnxblxWzNgmdqihXnJHFeVyzxMhsXzd
24 BEBaxeaPgQttvqXvHPPEOU1stPDeeuGfmgDkQcYpzQgvcCDKrcYwHF1VpZDEk
25 MyuPcvxtsWvgrbyvK0nbeQhnguXHnyj)jw5if1aw+AMBzScsoSumwPssjCP1LLBPF1Gt1Dp2ZmE
26 jarrjufhgxdzywosrb1PAsrvOpZlaLbDQgQzvzOvHrStPp1b111jVvvwvhpnVzeWVhmp3
27 KMVcdelzZXTwocGKhcncozRSdwEhpeNfJrJlwCvkrVzJcZIYFPcyrJxjOpKkhVucCugE
28 luwLBcmqzWdvUPUBRjZjhwEXXHsvBijqVVIEGRWRSHPeKUJcpipssVmUmcDzZpche2myRtJkyXk
29 oLEMjtaCsKmNOKNhygTlxKhaSifidWNBzYYHCGMtjg1tSlzWjz0AMuhmWvincBydQMDpaclReYU
30 SeEnkclzGopkBozDQfXxQgHrV1AqsiJCsrauawyskboAnjctftReVr1BFBK1LSPGmrLsrn2Kf0U
31 wnCYamNKNhadsGMXwBaeDrMx0NeuSuaSc1wgtgavosvusBwqOLvkNnsRaaPwrHtdwJGivXvfd
32 sxbXzJwjaITTpPwogOnPwcbhfzBlzKEBUECMLUKQrvVwgluodKrhjJmluoAhiTKVkcipM7J
33 AY1VsriBTLswuyYwCuuawoSrlwXJNLAflzrZXBEOCTtNptdJkrlfhuPnysjPjeWMDcfQdEs1g
34 feiJbjrkofNLjkiIgblfYR1ldJpjEdsdJzreCqGUh1IkjfwrrfjPwsASxnsqHKAMJIPuOHyZxui
35 THBokV2UgmalVBlgVpgEGpeliDfzYKhmAmJlwblwQHwesLuvQrUWEoCtwdkalMnyVvZfzomxyhX
36 LhFySwiPrhmVHQGJFzWSGUiKaYthUTSLwsobkBDpVJbhUxzuNrlPKzVtNoWkrngtEkazGaWNlfr
37 RdYbWmbtMtygloTyNvFyvvtotfTqfamVSmkApgbgyffxfsgNqkxTriegbsaChypYNUqcFxYEBgslzqM
41 leH1HfOdvhwFLIGaleFxLllgkAkJehM2LOROOArPPNhiVfPvPrUPqmV2sJfhpRpHoyvkaLuz
```

← → C hub.gke2.mybinder.org/user/python-ipython-in-depth-h1p6ys8/notebooks/binder/Untitled1.ipynb?kernel_name=python3#

jupyter Untitled1 最新检查点: 3分钟前 (已自动保存)

File Edit View Insert Cell Kernel Widgets Help 可信 Python 3 O

运行 代码 Download GitHub Binder Memory:

```
In [1]: import re
strings='''<!--
k4ewtloYgcFqaJNhHVGxXDiQmzjfcpYbzxlWrVcqsmUbCunkfxzWdzjUZMiGghRRIUwGmYmvnJ1HEmbT
MUKLECKdCtheSYBpIElRnZugFAXDRtQPpceBgbfaRVvvguRLvxkAdLoeCKxsDuvBCwdpMMwuElEg
ENihrpCLhujoBgPDRPvzcrwadMMbkmkzCCzotPfDrizBqMblmxrXhNlCufprWXxHSpplkoLcrJd
vuyFCZtqKLHwIyzoXeglkxhVJImeuySguVmTCyMshQtvzpw1bd0HnbauwvUYCmgznObgByPw
TDQheAbsaMljTmaOKmNsLziVmEnfXqdATQ1jtwtcyHeMwQTnxblxWzNgmdqihXnJHFeVyzxMhsXzd
BEBaxeaPgQttvqXvHPPEOU1stPDeeuGfmgDkQcYpzQgvcCDKrcYwHF1VpZDEk
MyuPcvxtsWvgrbyvK0nbeQhnguXHnyj)jw5if1aw+AMBzScsoSumwPssjCP1LLBPF1Gt1Dp2ZmE
jarrjufhgxdzywosrb1PAsrvOpZlaLbDQgQzvzOvHrStPp1b111jVvvwvhpnVzeWVhmp3
KMVcdelzZXTwocGKhcncozRSdwEhpeNfJrJlwCvkrVzJcZIYFPcyrJxjOpKkhVucCugE
luwLBcmqzWdvUPUBRjZjhwEXXHsvBijqVVIEGRWRSHPeKUJcpipssVmUmcDzZpche2myRtJkyXk
oLEMjtaCsKmNOKNhygTlxKhaSifidWNBzYYHCGMtjg1tSlzWjz0AMuhmWvincBydQMDpaclReYU
SeEnkclzGopkBozDQfXxQgHrV1AqsiJCsrauawyskboAnjctftReVr1BFBK1LSPGmrLsrn2Kf0U
wnCYamNKNhadsGMXwBaeDrMx0NeuSuaSc1wgtgavosvusBwqOLvkNnsRaaPwrHtdwJGivXvfd
sxbXzJwjaITTpPwogOnPwcbhfzBlzKEBUECMLUKQrvVwgluodKrhjJmluoAhiTKVkcipM7J
AY1VsriBTLswuyYwCuuawoSrlwXJNLAflzrZXBEOCTtNptdJkrlfhuPnysjPjeWMDcfQdEs1g
feiJbjrkofNLjkiIgblfYR1ldJpjEdsdJzreCqGUh1IkjfwrrfjPwsASxnsqHKAMJIPuOHyZxui
THBokV2UgmalVBlgVpgEGpeliDfzYKhmAmJlwblwQHwesLuvQrUWEoCtwdkalMnyVvZfzomxyhX
LhFySwiPrhmVHQGJFzWSGUiKaYthUTSLwsobkBDpVJbhUxzuNrlPKzVtNoWkrngtEkazGaWNlfr
RdYbWmbtMtygloTyNvFyvvtotfTqfamVSmkApgbgyffxfsgNqkxTriegbsaChypYNUqcFxYEBgslzqM
leH1HfOdvhwFLIGaleFxLllgkAkJehM2LOROOArPPNhiVfPvPrUPqmV2sJfhpRpHoyvkaLuz
```

← → C 不安全 | pythonchallenge.com/pcc/def/linkedlist.html

linkedlist.php



Level 4

How to pass the customs: This level started with only one picture, without any other prompts, but when I clicked on the picture, it went to another webpage, which only had a sentence "and the next nothing is 44827", so I put 44827 into nothing in the URL and started the jump again. Or just "and the next nothing is 45439," the five-digit number that should be the entry to the next jump. Then I jumped around again and again, and I tried it a couple of times and it didn't seem right, so I went to the source code.

Solution: Follow the prompts above the web source code, which gives us a library urllib and then tells us not to try everything because it is infinite and 400 loops are plenty.

Pass through pictures:



```
< → C 不安全 | view-source:www.pythonchallenge.com/pc/def/linkedlist.php  
自动换行 □  
1 <html>  
2 <head>  
3 <title>follow the chain</title>  
4 <link rel="stylesheet" type="text/css" href="../style.css">  
5 </head>  
6 <body>  
7 <p>urllib may help. DON'T TRY ALL NOTHINGS, since it will never  
8 end. 400 times is more than enough. -->  
9 <center>  
10 <a href="linkedlist.php?nothing=12345"></a>  
11 <br><br><font color="gold"></center>  
12 Solutions to previous levels: <a href="http://wiki.pythonchallenge.com/">Python Challenge wiki</a>.  
13 <br><br>  
14 IRC: irc.freenode.net #pythonchallenge  
15 </body>  
16 </html>  
17  
18
```

hub.gke2.mybinder.org/user/ipython-ipython-in-depth-fitlgoqqz/notebooks/binder/Untitled1.ipynb?kernel_name=python3#

jupyter Untitled1 最新检查点: 5分钟前 (已自动保存)

File Edit View Insert Cell Kernel Widgets Help 可信 Python 3 Memory:

运行 代码 Download GitHub Binder

```
In [1]: import requests  
import re  
import time  
url = "http://www.pythonchallenge.com/pc/def/linkedlist.php?nothing="  
key = "90990"  
pattern = r'and the next nothing is ([0-9]+)'  
while True:  
    print(url + key)  
    curtext = requests.get(url + key).text  
    print(curtext)  
    if "Divide by two" in curtext: # handle "Divide by two"  
        key = str(int(key)/2)  
        continue  
    if "and the next nothing" not in curtext:  
        break  
    key = re.search(pattern,curtext).group(1)  
  
http://www.pythonchallenge.com/pc/def/linkedlist.php?nothing=90990  
and the next nothing is 27935  
http://www.pythonchallenge.com/pc/def/linkedlist.php?nothing=27935  
and the next nothing is 99927  
http://www.pythonchallenge.com/pc/def/linkedlist.php?nothing=99927  
and the next nothing is 41785  
http://www.pythonchallenge.com/pc/def/linkedlist.php?nothing=41785  
and the next nothing is 32660  
http://www.pythonchallenge.com/pc/def/linkedlist.php?nothing=32660  
and the next nothing is 4328  
http://www.pythonchallenge.com/pc/def/linkedlist.php?nothing=4328  
and the next nothing is 42067  
http://www.pythonchallenge.com/pc/def/linkedlist.php?nothing=42067  
and the next nothing is 8743  
http://www.pythonchallenge.com/pc/def/linkedlist.php?nothing=8743
```

http://www.pythonchallenge.com/pc/def/peak.html

顶峰地狱 - http://www.pythonchallenge.com/pc/def/peak.html 切换到这个标签页

http://www.pythonchallenge.com/pc/def/peak.html - Google 搜索

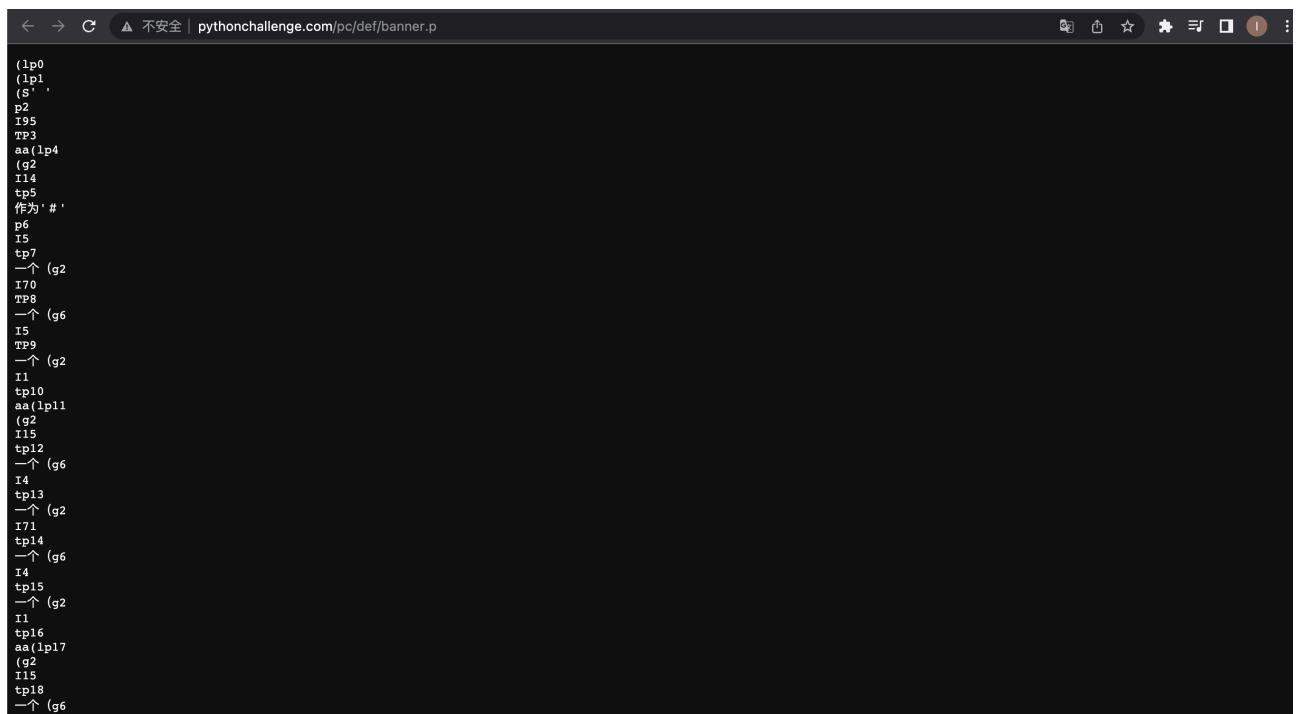
```
<html>  
<head>  
<title>  
<link >  
</head>  
<body>  
<p>urllib may help. DON'T TRY ALL NOTHINGS, SINCE IT WILL NEVER  
end. 400 times is more than enough. -->  
<center>  
<a href="linkedlist.php?nothing=12345"></a>  
<br><br><font color="gold"></center>  
Solutions to previous levels: <a href="http://wiki.pythonchallenge.com/">Python Challenge wiki</a>.  
<br><br>  
IRC: irc.freenode.net #pythonchallenge  
</body>  
</html>
```

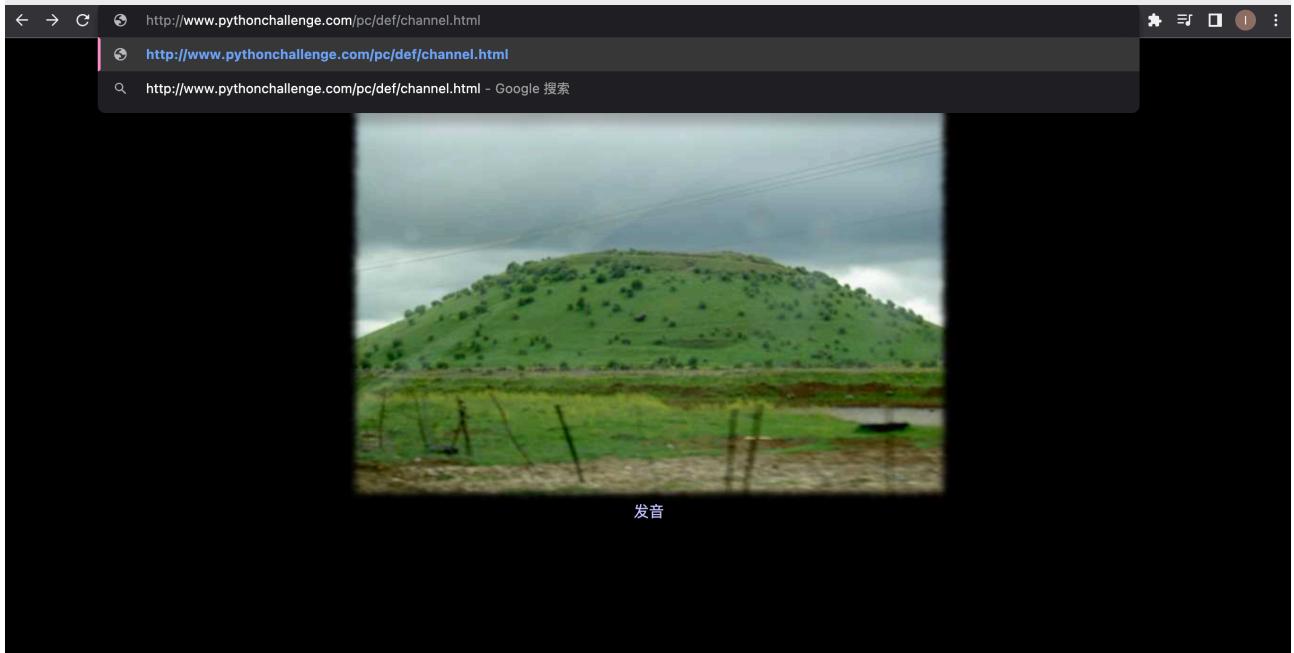
Level 5

How to pass the customs: This page has nothing but a picture and a "pronounce it". I don't know what to do with this. So I chose to continue the idea of the previous levels and look at the source code first. 'peak hell sounds familiar? 'and then entered the <http://www.pythonchallenge.com/pc/def/pickle.html>. But that still doesn't solve the problem. Then I looked at the source again, and there was a peakhell tag, and inside it was banner.p and it opened up to a bunch of random text.

Solution: First deserialize the contents of banner.p. After deserialization, this is a two-dimensional array. Look for the pattern and see that the sum of the numbers in each line is 95, and that each tuple in it consists of two parts, one character (space or #) and one number, such as: (" ", 14), ("#", 5). If you think of it as a matrix, the number in the tuple represents the number of previous character repeats, and each line has 95 characters.

Pass through pictures:





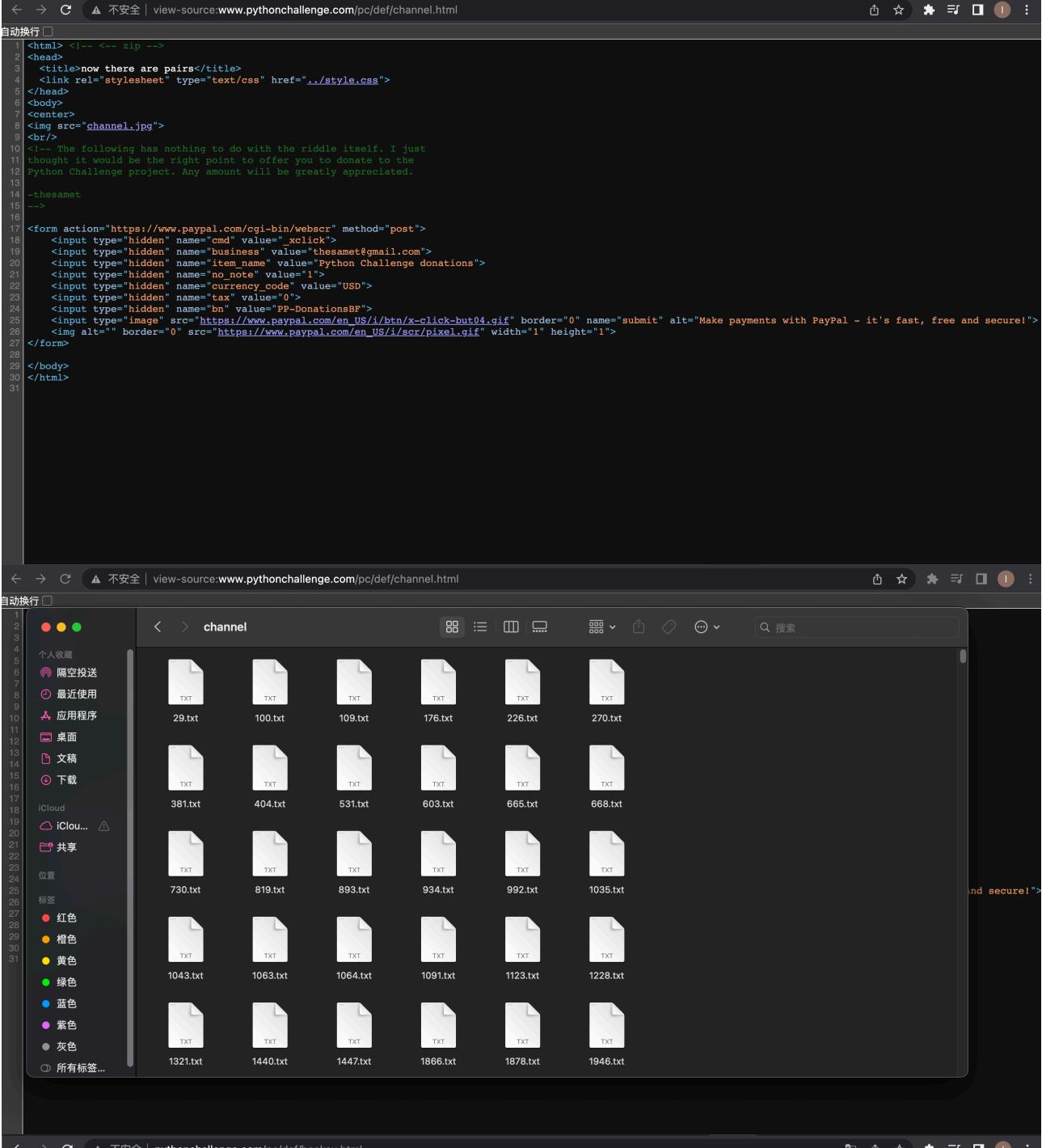
Level 6

How to pass the customs: Open the website, found only pictures and titles, I did not see what useful information, so look at the source code. Found a <! - < - zip - > more strange, played so few close and find place to look for clues is basic it is abnormal, the zip is a file extension, so try to change an address at www.pythontechchallenge.com/pc/def/channel.zip.

Solution: Change html to zip, download a zip file, open the file will find, all are some txt files. The document says: Next nothing is 91038, similar to the previous fourth level. There's a readme.txt in there: it prompts us to start with 90052.txt, and the answers are in the zip pack! According to the thoughts of the fourth level, the result 46145.txt will be printed: Collect the comments. Combined with the answer just prompted in the zip package, the original answer is the annotation of each file

combined. I print all the file notes and get the result: hockey. Open open connection: http://www.pythonchallenge.com/pc/def/hockey.html. You get the following sentence: it's in the air. look at the letters.

Pass through pictures:

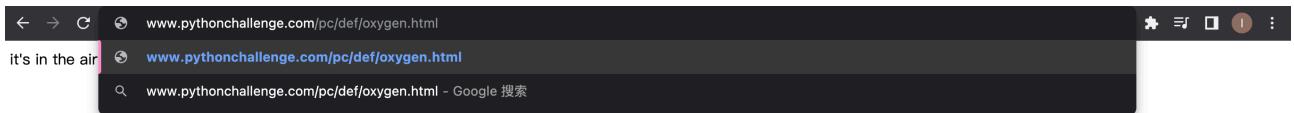


```
<!-- zip -->
<head>
<title>now there are pairs</title>
<link rel="stylesheet" type="text/css" href="../style.css">
</head>
<body>
<center>

<br/>
<!-- 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
-->
<form action="https://www.paypal.com/cgi-bin/webscr" method="post">
<input type="hidden" name="cmd" value="_xclick">
<input type="hidden" name="business" value="thesamet@gmail.com">
<input type="hidden" name="item_name" value="Python Challenge donations">
<input type="hidden" name="no_note" value="1">
<input type="hidden" name="currency_code" value="USD">
<input type="hidden" name="tax" value="0">
<input type="hidden" name="bn" value="PP-DonationsBF">
<input type="image" alt="https://www.paypal.com/en_US/i/btn/x-click-but04.gif" border="0" name="submit" alt="Make payments with PayPal - it's fast, free and secure!">

</form>
</body>
</html>
```

it's in the air. look at the letters.

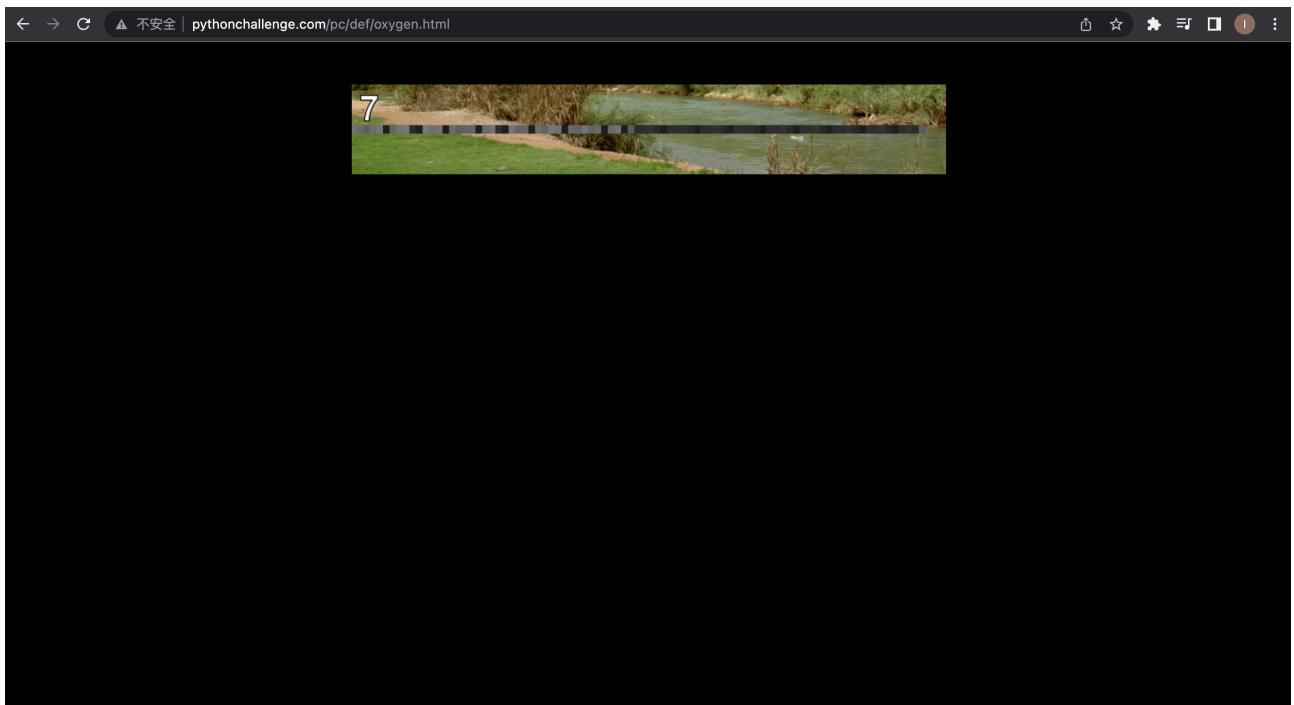


Level 7

How to pass the customs: First log in to the seventh question of the website, open the discovery of a picture, in accordance with the previous ideas to directly view the source page. After viewing the source code is not useful, so think will not answer in the picture, so ready to deal with the picture in that "black and white gray" bar code.

Solution: It is found that after the image is processed by PIL, R,G and B in the code are the same. Visually, it may be ASCII code. Output integrity through python.

Pass through pictures:



hub.gke2.mybinder.org/user/ipython-ipython-in-depth-zkf1d2io/notebooks/binder/Untitled5.ipynb?kernel_name=python3#

jupyter Untitled5 最新检查点: 9 分钟前 (已自动保存)

File Edit View Insert Cell Kernel Widgets Help

运行 Download GitHub Binder Memory:

```
In [1]: from urllib.request import urlretrieve
from PIL import Image
import sys

urlretrieve("http://www.pythonchallenge.com/pc/def/oxygen.png", "oxygen.png")
im = Image.open("oxygen.png")
width, height = im.size
pic = im.load()
h = height // 2
for x in range(width):
    print(pic[x,h])
(97, 97, 97, 255)
(97, 97, 97, 255)
(97, 97, 97, 255)
(97, 97, 97, 255)
(97, 97, 97, 255)
(97, 97, 97, 255)
(97, 97, 97, 255)
(97, 97, 97, 255)
(114, 114, 114, 255)
(114, 114, 114, 255)
(114, 114, 114, 255)
(114, 114, 114, 255)
(114, 114, 114, 255)
(114, 114, 114, 255)
(114, 114, 114, 255)
(114, 114, 114, 255)
(116, 116, 116, 255)
(116, 116, 116, 255)
(116, 116, 116, 255)
(116, 116, 116, 255)
(116, 116, 116, 255)
(116, 116, 116, 255)
```

hub.gke2.mybinder.org/user/ipython-ipython-in-depth-zkf1d2io/notebooks/binder/Untitled5.ipynb?kernel_name=python3#

jupyter Untitled5 最新检查点: 9 分钟前 (已自动保存)

File Edit View Insert Cell Kernel Widgets Help

运行 Download GitHub Binder Memory:

```
In [1]: from PIL import Image
import sys

urlretrieve("http://www.pythonchallenge.com/pc/def/oxygen.png", "oxygen.png")
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
smart guy, you made it. the next level is [105, 10, 16, 101, 103, 14, 105, 16, 121]

In [3]: answer = [105, 110, 116, 101, 103, 114, 105, 116, 121]
for l in answer:
    print(chr(l), end='')

integrity
```

