

攻略作者: FIREDOM

GitHub: <https://github.com/firedom>

本来是不想写攻略的, 直到我不小心找到了做这个发布页面并看到了这个说明:



发不发答案是我的自由, 凭什么不让发答案?

于是我把我的过关过程直接写了出来, 顺便带上每关的地址, 方便下次玩的时候接关=v=

网页解谜游戏

地址:

<https://harrynull.tech/cipher/>

输入名字, 按start开始

Stage1:

<https://harrynull.tech/cipher/link.html>

点击链接进入第二关。

Stage2:

<https://harrynull.tech/cipher/missinglink.html>

Inspect Element查看网页源代码, 找到隐藏链接:

```
23 <p>
24 点击链接进入第三关.....诶? 链接呢?
25 </p>
26 <p style="color: #000033;">
27 <br> <!--这里换行是为了避免跟上一关链接位置一样啦-->
28 链接在这里哦-><a href="thethirdstage.html" style="color: #000033;">第三关</a>
29 </p>
30 <iframe src="rank_frame.php" class="topframe"></iframe>
31 </body>
32 </html>
33
```

Stage3:

<https://harrynull.tech/cipher/thethirdstage.html>

第三关提示时往上看和URL。

```
6 <title>第三关 - Universal Resource Locator</title>
7 <link href="css/style.css" rel="stylesheet" type="text/css">
8 <script>
9   (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
10     (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
11     m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
12   })(window,document,'script','https://www.google-analytics.com/analytics.js','ga');
13
14   ga('create', 'UA-100793414-1', 'auto');
15   ga('send', 'pageview');
16 </script>
17 </head>
18
19 <body>
20 <h1>
21 Stage 3
22 </h1>
23 <p>
24 这次真的没有链接了。怎么办呢?
25 </p>
26 <p style="color: #000033;">
27 都说了没有链接了.....别往这里看, 往上面看!
28 </p><iframe src="rank_frame.php" class="topframe"></iframe>
29 </body>
30 </html>
31
```

于是试试修改链接:

<https://harrynull.tech/cipher/thefourthstage.html>

Stage4:

网页源码最后有提示:

```
28 </p>
29 <iframe src="rank_frame.php" class="topframe">
30 </body>
31 </html>
32 <!--
33 这里就是传说中的源代码! 恭喜你。
34 下一关是ultimate.html
35 -->
```

于是得到答案

<https://harrynull.tech/cipher/ultimate.html>

Stage5:

<https://harrynull.tech/cipher/ultimate.html>

提示是“The Answer to the Ultimate Question of Life, The Universe, and Everything =”，百度这个提示可知答案是42。输入42并按确定。

Stage6:

<https://harrynull.tech/cipher/whoisshe.html>

Google 搜索提示图片，并打开其中一个搜索结果：

<https://www.animecharactersdatabase.com/source.php?id=2363>

这个角色的名字是Yuki Minakami，于是输入这个名字过关。

Stage7:

<https://harrynull.tech/cipher/blank.html>

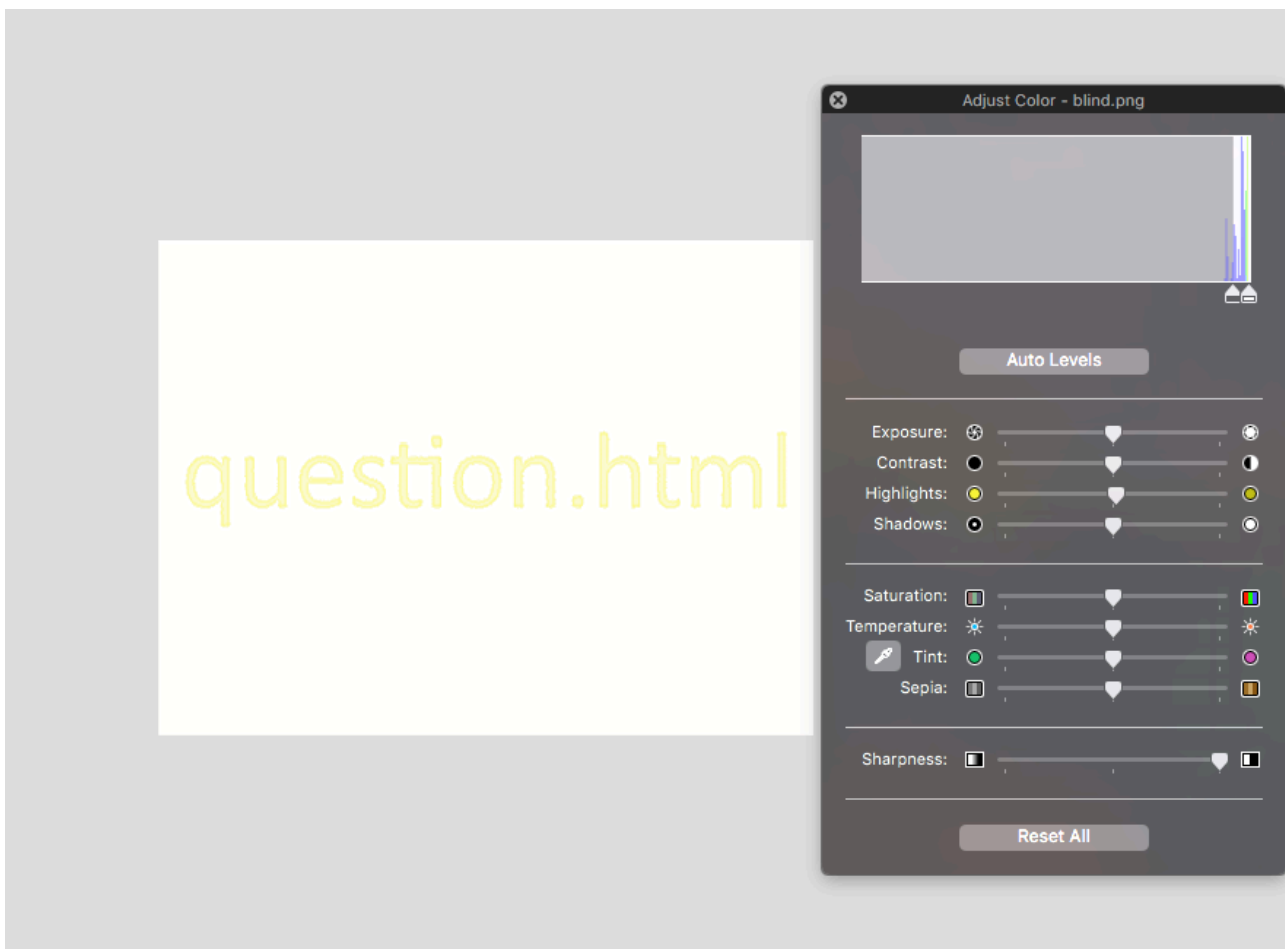
查看源代码看到有一个图片链接：

```
17 </head>
18
19 <body>
20 <h1>
21 Stage 7
22 </h1>
23 <p>
24 
25 </p>
26 <iframe src="rank_frame.php" class="topframe"><
27 </body>
28 </html>
```

下载这个图片并用16进制查看器找线索。没有有用的提示。

直接打开图片发现有淡黄色痕迹，处理一下图像得到答案。

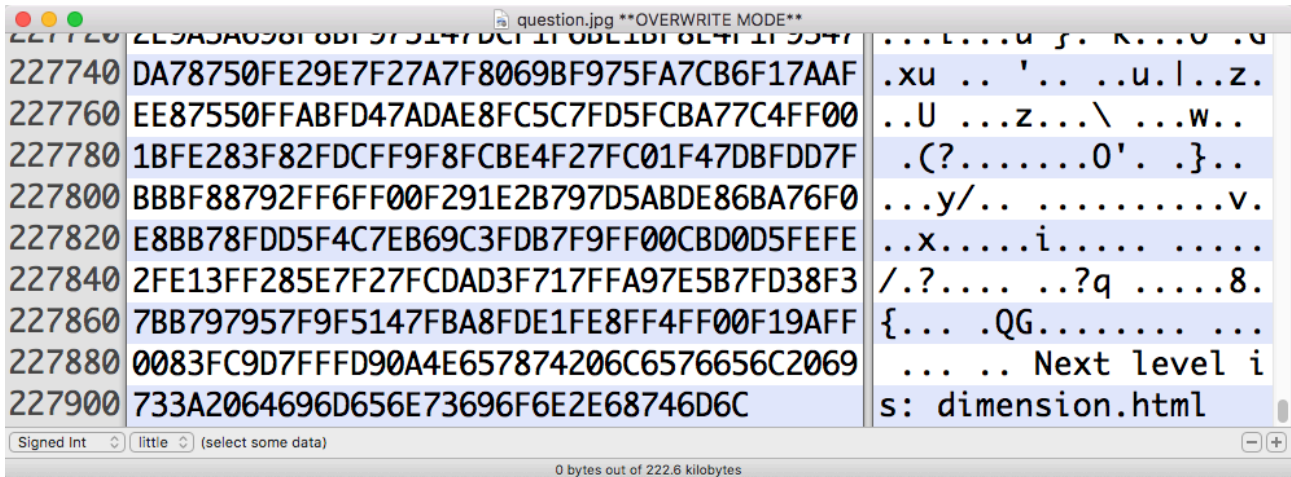
<https://harrynull.tech/cipher/question.html>



Stage8:

<https://harrynull.tech/cipher/question.html>

下载图片用16进制查看器打开:



得到答案: <https://harrynull.tech/cipher/dimension.html>

Stage9:

<https://harrynull.tech/cipher/dimension.html>

源码提示“不小心把密码留给你了, 可是.....好像有别的什么东西错了。是什么呢?”

尝试各种密码:

password

password 10

Stage 10

都不对, 于是看源码

<h1>

Stage 9

</h1>

<p>

请输入密码:

<form action="check.php" method="POST">

<input type="input" name="password" size=20 maxlength=20 value="password">

<input type="hidden" name="stage" value="0">

<input type="submit" value="确定"></form>

</p>

<iframe src="rank_frame.php" class="topframe"></iframe>

</body>

</html>

<h1>

Stage 6

</h1>

```

<p>
 <br>
她是谁呢?
<form action="check.php" method="POST">
<input type="input" name="password" size=4 maxlength=4>
<input type="hidden" name="stage" value="6">
<input type="submit" value="确定"></form>
<iframe src="rank_frame.php" class="topframe"></iframe>
</p>
</body>
</html>

<h1>
Stage 5
</h1>
<p>
The Answer to the Ultimate Question of Life, The Universe, and Everything =
<form action="check.php" method="POST">
<input type="input" name="password" size=20 maxlength=20>
<input type="hidden" name="stage" value="5">
<input type="submit" value="确定"></form>
<iframe src="rank_frame.php" class="topframe"></iframe>
</p>
</body>
</html>

```

找不同发现hidden type 后面的value等于这关的编号。而第9关这里等于0。于是修改value=9并回车，密码不需要改。

Stage10:

<https://harrynull.tech/cipher/gonetoofar.html>

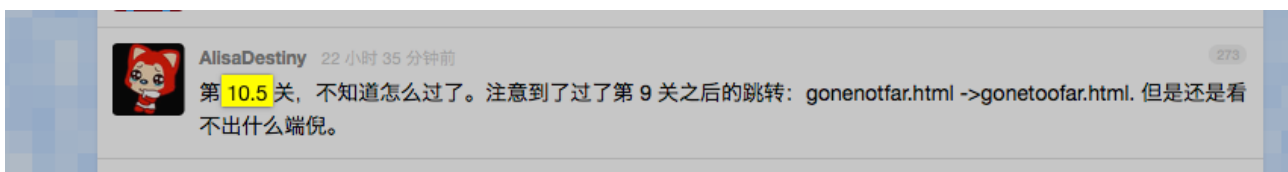
既然走远了那就试试gotoright.html

然而页面提示找不到对象=3=

gonetoofar-1.html 也没有对象

这个时候要去看看游戏发布地址的留言。搜索10.5，并在第三页找到了线索：

<https://www.v2ex.com/t/382225?p=3>



输入：<https://harrynull.tech/cipher/gonenotfar.html>

确实又跳回10.5页面了，留言没说谎。

既然会自动跳转，那我用curl下载下来看就不会自动跳转了。

Terminal执行curl -O <https://harrynull.tech/cipher/gonenotfar.html>

在文件末尾发现提示：

```
<!--
```

这里才是真正的第十关！

下一关是message.html

```
-->
```

密文: c10f77963a2b21079156a0e5c5a4bb3c

类型: 自动

[\[帮助\]](#)

查询

加密

查询结果:
digest[\[添加备注\]](#)[md5解密记录](#)[收藏本站](#) [返回首页](#) [破解记录返回首页 ...](#)[pmd5.com/about/histo.....](#) - 百度快照[unit xsdauerrv.dfm Source View - Source code download - Three...](#)

Stage11:

<https://harrynull.tech/cipher/message.html>试试<https://harrynull.tech/cipher/c10f77963a2b21079156a0e5c5a4bb3c>

然而并没有对象。

那就试试百度，搜索这个：

然而第一个搜索结果打开特别慢。既然有可能是md5，那就找个打开速度快的md5解密网站解密。

搜索md5解密并打开第一个搜索结果：

输入提示字符串，得到结果digest

等了半天才有反应的第一个页面的解谜结果也是digest

<http://pmd5.com/about/detail.aspx?code=c10f77963a2b21079156a0e5c5a4bb3c>

Stage 12

不正确的字符编码引起了网页的乱码。这一关的密码是世界上第一台通用计算机的英语缩写。

确定

于是输入地址<https://harrynull.tech/cipher/digest.html> 过关

Stage12:

<https://harrynull.tech/cipher/digest.html>

乱码看不懂，试试换个编码。

换成UTF-8就不是乱码了。

百度“世界上第一台通用计算机的英语缩写”，结果是ENIAC。

输入ENIAC成功跳到13关

Stage13:

<https://harrynull.tech/cipher/snownoise.html>

快速放大缩小图片能看到一个雪花，但是输入snowflower或者snow并不是答案。

再去发布页面搜索留言，在<https://www.v2ex.com/t/382225?p=2> 找到了提示：



但是这个网页只有一张图片哪来的第二张？看了一下图片地址：

<https://harrynull.tech/cipher/img/snowa.png>

是snowa，那是不是还有snowb&cdefgh，试试看～

测试发现snowb存在

<https://harrynull.tech/cipher/img/snowb.png>

snowc~h和snow不存在。

那应该就是两张图片了吧。

手动选中这两张图片并手动快速切换会看到文字interference，于是输入地址：

<https://harrynull.tech/cipher/interference.html>

过关。

Stage14:

<https://harrynull.tech/cipher/interference.html>

一共有四个提示：

第十四关 - 定律

SN3W3IS

<!--

提示1：umop apisdn

提示2：公司？仅此而已？

-->

如果是公司的话umop对应comp，api和sdn都有其意义。

百度SN3W3IS找到<http://it007.com/thread-815050-9-1.html>



西门子的英文是SIEMENS，于是很容易发现：

SN3W3IS -> 上下&左右镜像文字 ->



SIEMENS

再按这个套路对待提示1，得到：

NDSIdv dOWN

Ups!d

以上当作密码都不对，直接百度umop apisdn

<http://www.urbandictionary.com/define.php?term=umop%20apisdn>

upside down也不对

发布页面有答案<https://www.v2ex.com/t/382225?p=4>



果然答案是“欧姆”

Stage15:

<https://harrynull.tech/cipher/PNGsNot.html>

源码提示：

<!--

文件头？那是什么？

-->

下载网页里的图片用16紧致查看器打开发现文件头跟PNG的文件头不一样。百度png89a发现是GIF的文件头。修改文件头并打开这个图片找到答案。

Stage16

<https://harrynull.tech/cipher/flatland.html>

好像是3D画，将图片全屏显示名与远离显示器看到文字blocky.html（离太近看不清）

Stage17:

<https://harrynull.tech/cipher/blocky.html>

提示：

19b31878f1200

Who am I?

<!--世界上有10种人.....-->

第十七关 - 七七四十九

第一行提示转为二进制的话是：

0x19b31878f1200: 0001 1001 1011 0011 0001 1000 0111 1000 1111 0001 0010 0000 0000

10进制：

452112006713856

提示七七四十九，那么 $\sqrt{49}=7$

于是 $\sqrt{452112006713856} = 21262925.638628753855674$

然而这个结果不是答案。

1695也不是

继续去发布页找线索：

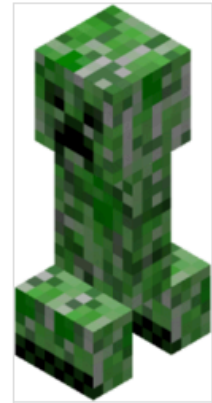


可能是找在MC里谁最像这个图形。去百度百科发现应该是这个：

西久伯歌

生命：20

当闪电劈中普通爬行者附近 3~4 个方块时，会生成高压爬行者。它的爆炸强度是一般的爬行者的 2 倍。高压爬行者有着明显的蓝色电弧围绕。爬行者、骷髅、凋灵骷髅、僵尸被高压爬行者击杀会掉落他们的头颅。



爬行者

<https://harrynull.tech/cipher/obstacle.html>

Obstacle n. 障碍 (物); 障碍物 (绊脚石, 障碍栅栏);

这一关的密码是：helloworld

goodbyeworld.html也不对

<https://harrynull.tech/cipher/readable.html>

[illegible]

太乱了看不清，继续整理：



百度“static char test_bits[] = { 0x00, 0x80, 0x00, 0x00, 0x00, 0x15, 0x80, 0xA0, 0x44, 0x80, 0x2B, 0xA9, 0x66, 0xE4, 0x80, 0xB9, 0xAA, 0x28, 0x4C, 0x96, 0x09, 0xB3, 0x2C, 0x54, 0xAA, 0x31, 0x82, 0x2A, 0x95, 0xAA, 0x80, 0x01, 0x0C, 0x00, 0x00, };”才知道是X BitMap。XBM file https://en.wikipedia.org/wiki/X_BitMap

用ffmpeg转换：

ffmpeg -i 233.xbm out.png

转换结果：



Stage20:

<https://harrynull.tech/cipher/regular.html>

提示：

第二十关 - 普通的表达式我们普通地摇

T29wcy4gWW91IHRyYXBwZWwW91cnNlbGYuIEkgZ3Vlc3MgdGhhdCdzIGl0IHRoZW4uIFRoYW5r
cyBmb3lmdGVzdGluZy4gWW91IG1heSBhcyB3ZWxslGxpZSBkb3dulGFuZCBnZXQgYWNjbGltYX
RIZCB0byB0aGUgYmVpbmcgZGVhZCBwb3NpdGlvbi4iCJJJ20ga2lkZGluZy4gTm90IGFib3V0IH
vdSB0cmFwcGluZyB5b3Vyc2VsZiWgdGhvdWdoLiBUaGF0IHJlYWxseSB0YXBwZW5lZC4gSGVvZ
SwgSSdsbCBsb3dlciB0aGUgZ2xhc3MulEdvIG9uLi4uIEZpbmlzaCB0aGUgdGVzdC5UMjl3Y3k0Z
1dXOTFJSFJ5WVhCd1pXUWdlzkxY25ObGJHwXVJRWtnWjNWbGMzTWdkR2hoZENkeklHbDBJ
SFJvWlc0dUIGUm9ZVzVyY3lCbWlzSWdkR12Z6ZEdsdVp5NGdXVzkxSUcxaGVtQmhjeUlzWld4c
0IHcBaU0JrYjNkdUIHRnVaQ0JuWlhRZ1lXTmpiR2x0WVhSbFpDQjBieUlWYUdVZ1ltVnBibWNnWk
dWafpDQndiM05wZEdsdmJpNGlJQ0pKSjlwZ2EybgtaR2x1Wnk0Z1RtOTBJR0ZpYjNWMEllbHZk
U0lwY21Gd2NHHVaeUl1YjNeWMyVnNaaXdnZEdodmRXZG9MaUJvYUdGMEllSmxZV3hzZVNCb
1IYQndaVVsWkM0Z1NHVnlaU3dnU1Nkc2JDQnNiM2RsY2lCMGFHVWdaMnhoYzNNdUIFZHJRZl
1TGk0dUIFWnBibWx6YUNCMGFHVWdkR1Z6EM0PQ

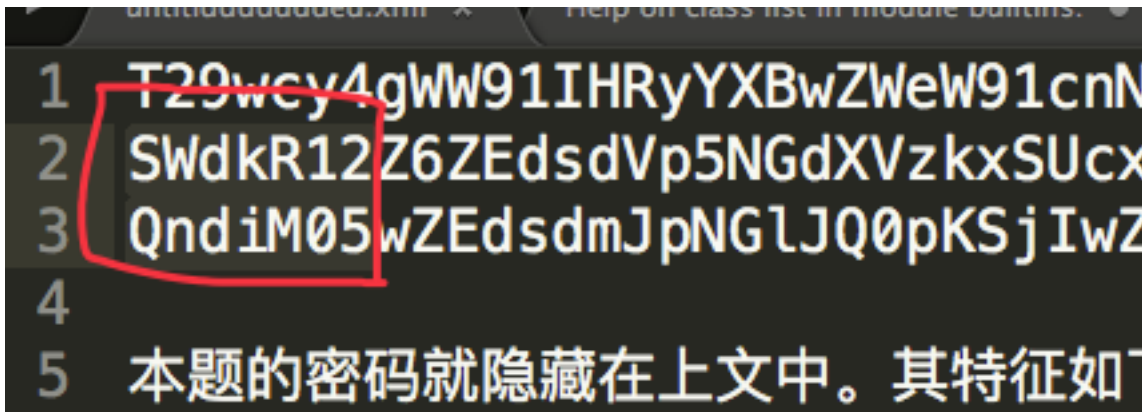
本题的密码就隐藏在上文中。其特征如下：

开头和结尾均为大写字母；

第3、4位是小写字母；

第6、7位是数字；
倒数第三位与最后一位相同。

用正则表达式：
[A-Z][A-Za-z0-9][a-z][a-z][A-Za-z0-9][0-9][0-9]
搜到了前7个字符符合要求的字符串：



再根据

剩下的条件判断，密码只可能是SWdkR12Z6Z
验证后成功进入下一关。

Stage21:
<https://harrynull.tech/cipher/header.php>

提示：

第二十一关 - 一片虚无

<h1>

Stage 21

</h1>

<p>Void</p>

<iframe src="rank_frame.php" class="topframe"></iframe>

</body>

</html> <!-- .php? -->

可能是后边有php代码但是注释掉了，先试试扩展名改成html,结果提示找不到对象。

Void.php和Void.html也是一样。

User-Agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/603.3.8 (KHTML, like Gecko) Version/10.1.2 Safari/603.3.8
▼ Response Headers	
Name	Value
Server	Apache/2.4.18 (Ubuntu)
Content-Type	text/html; charset=UTF-8
Date	Tue, 15 Aug 2017 12:20:53 GMT
Connection	Keep-Alive
Content-Length	584
Vary	Accept-Encoding
Keep-Alive	timeout=5, max=100
Content-Encoding	gzip
Next-Level	loginfirst.html

看了一下除源码意外的信息，并找到了下一关的地址。

Stage22:

<https://harrynull.tech/cipher/loginfirst.html>

看源代码发现密码

```

22 </hi>
23 <p>
24 请先登录
25 <form action="check.php" method="POST">
26 <p>用户名<span style="width:2em;display:inline-block"></span><input type="input" name="username" value="admin" size=20 maxlength=20></p>
27 <p>密码<span style="width:3em;display:inline-block"></span><input type="password" name="password" value="admin_key" size=20 maxlength=20></p>
28 <p>重复密码<span style="width:1em;display:inline-block"></span><input type="password" name="passwordr" size=20 maxlength=20></p>
29 <input type="hidden" name="stage" value="22">
30 <input type="submit" value="登录"></form>
31 </p>
32 <iframe src="rank_frame.php" class="topframe"></iframe>
33 </body>
34 </html>

```

输入admin_key并回车。

Stage23:

https://harrynull.tech/cipher/admin_page.html

好像需要输入密码，但是并没有地方可以输入。

看源码发现密码输入框那里多了个hidden

```
22 </nl>
23 <p>
24 利用管理员模式通过这一关
25 <form action="check.php" method="POST">
26 <input type="input" name="password" value="admin" hidden size=20 maxlength=20>
27 <input type="hidden" name="stage" value="23">
28 <input type="submit" value="以管理员模式通关"></form>
29 </p>
30 <iframe src="rank_frame.php" class="topframe"></iframe>
31 </body>
32 </html>
```

去掉之后出现密码框

A screenshot of a web page titled "Stage 23" with a dark blue background. The text "利用管理员模式通过这一关" (Use administrator mode to pass this level) is centered. Below it is a login form with a text input field containing the word "admin" and a grey button labeled "以管理员模式通关" (Pass with administrator mode).

但是密码并不是admin

A screenshot of a web page with a dark blue background. It displays the message "没有管理员权限（给管理员吃个饼干试试？）" (No administrator privileges (try giving the administrator a cookie?)). At the bottom, there is a blue underlined link that says "<<返回上一页" (Return to the previous page).

意思是要改cookie么？

Name	Value	Domain	Path	Expires	Size	HTTP
visitflag	23	harrynull.tech	/cipher	12/9/2017, 2...	11 B	
playername	freedom	harrynull.tech	/cipher	12/7/2017, 8...	17 B	
isadmin	0	harrynull.tech	/cipher	Session	8 B	

改成1试试～

Name	Value	Domain	Path
isadmin	1	harrynu...	/
playername	123	harrynu...	/cipr
visitflag	34	harrynu...	/cipr

果然可以～（Safari好像不支持直接改cookie了，chrome还可以）

Stage24:

<https://harrynull.tech/cipher/numbergame.html>

提示：

第二十四关 - 数字游戏

feed = 2cfb

beef = 1o99

6422 = ____

<!-- 转换过程不损失信息 -->

<!-- NUMBERgame != STRINGgame -->

看发布页提示就是说要找任意进制转换器进行计算。

```
ans += DIALPAD[NUM][NUM][NUM][+ 1]
```

ppwangs 1天前

NUMBERgame != STRINGgame

feed = 可以转 16 进制，可以按照字符转，也可以按照单词转。

beef = 这个 o 可以视为 8 进制，但是无法串联，然后看成 25 进制楼上说不对，思路对。

24 题暂无思路。

SuperFashi 1天前 via Android

@ppwangs 写一个程序从 16-36 进制到 25-36 进制一个个试呗

U2FsdGVkX1 1天前

然后我需要一个进制转换器比如这个：
<https://github.com/cangli/python-tricky>

写测试代码并在解释器里运行：

```
def baseN(num,b):
    return ((num == 0) and "0" ) or ( baseN(num // b, b).lstrip("0") +
    "0123456789abcdefghijklmnopqrstuvwxyz"[num % b])

for x in range(10, 37):
    print(baseN(0xbeef, x), x)

for x in range(10, 37):
    print(baseN(0xfeed, x), x)
```

运行过程：

```
>>> def baseN(num,b):
...     return ((num == 0) and "0" ) or ( baseN(num // b, b).lstrip("0") +
...     "0123456789abcdefghijklmnopqrstuvwxyz"[num % b])
...
>>> for x in range(10, 37):
...     print(baseN(0xbeef, x), x)
...
48879 10
337a6 11
24353 12
1932c 13
13b55 14
e739 15
beef 16
9g24 17
86f9 18
727b 19
623j 20
55hc 21
4clh 22
4094 23
3ckf 24
3354 25
2k7p 26
2d19 27
269j 28
203e 29
1o99 30
1jqn 31
1fnf 32
1bt6 33
189l 34
14vj 35
11pr 36
>>> for x in range(10, 37):
...     print(baseN(0xfeed, x), x)
...
65261 10
45039 11
```

```

31925 12
23921 13
19ad7 14
1450b 15
feed 16
d4df 17
b37b 18
99ef 19
8331 20
70ke 21
62i9 22
588a 23
4h75 24
44ab 25
3ie1 26
38e2 27
2r6l 28
2jhb 29
2cfb 30
25s6 31
1vnd 32
1quk 33
1mff 34
1i9l 35
1ect 36
>>>

```

于是可知

```

feed(16) = 2cfb(30)
beef(16) = 1o99(30)

```

那答案应该是6422(10) =see(30)

```

>>> print(baseN(0x6422, 30))
see
>>>

```

Stage25:

<https://harrynull.tech/cipher/meowmeow.html>

提示:

第二十五关 - 喵喵

<!-- 不知道你有没有听说过一个叫jsfuck的东西 -->

<!-- 不要看到错误就慌了，继续探索说不定有新发现 -->

```

° ω / = / ` m` ) / ~       /*`∇ ` */ ['_']; o=(° -° ) =_3; c=(° Θ° ) =(° -° )-(° -° ); (° Д° ) =(° Θ° )=
(o^_^o)/(o^_^o);(° Д° )={° Θ° : '_', ° ω / : ((° ω /==3) +'_') [° Θ° ], ° -° /:(° ω /+ '_')[o^_^o -(° Θ° )], °
Д° /:(° -° ==3) +'_')[° -° ]}; (° Д° ) [° Θ° ] =((° ω /==3) +'_') [c^_^o];(° Д° ) ['c'] = ((° Д° )+ '_') [(° -° )+
(° -° )-(° Θ° )];(° Д° ) ['o'] = ((° Д° )+ '_') [° Θ° ];(° o° )=(° Д° ) ['c']+(° Д° ) ['o']+(° ω /+ '_')[° Θ° ]+ ((° ω /==3)
+ '_') [° -° ]+ ((° Д° )+ '_') [(° -° )+(° -° )]+ ((° -° ==3) +'_') [° Θ° ]+((° -° ==3) +'_') [(° -° )-(° Θ° )]+(° Д° ) ['c']
+((° Д° )+ '_') [(° -° )+(° -° )]+ (° Д° ) ['o']+(° -° ==3) +'_') [° Θ° ];(° Д° ) ['_'] =(o^_^o) [° o° ] [° o° ];(°
ε° )=((° -° ==3) +'_') [° Θ° ]+ (° Д° ) . Д° /+(° Д° )+ '_') [(° -° )+ (° -° )]+((° -° ==3) +'_') [o^_^o -° Θ° ]+

```

```
((' - ==3) + '_') [' ° ° ] + ( ' ω ° / + '_') [' ° ° ]; ( ' - ) += ( ' ° ° ); ( ' ° ° ) [' ° ° ] = '\\'; ( ' ° ° ). ° ° / = ( ' ° ° + ' - )
[o^_ ^o - ( ' ° ° )]; (o° - ° o) = ( ' ω ° / + '_') [c^_ ^o]; ( ' ° ° ) [' ° ° ] = '\\'; ( ' ° ° ) [' ° ° ] ( ( ' ° ° ) [' ° ° ] ( ' ° ° + ( ' ° ° ) [' ° ° ] + ( '
° ° ) [' ° ° ] + ((o^_ ^o) + (o^_ ^o)) + ((o^_ ^o) + (o^_ ^o)) + ( ' ° ° ) [' ° ° ] + ((o^_ ^o) + (o^_ ^o)) + ( ' - ) + ( ' ° ° ) [' ° ° ]
+ ((o^_ ^o) + (o^_ ^o)) + ((o^_ ^o) + (o^_ ^o)) + ( ' ° ° ) [' ° ° ] + ((o^_ ^o) + (o^_ ^o)) + ( ' - ) + ( ' ° ° ) + ( ' ° ° ) [' ° ° ]
ε° ] + ((o^_ ^o) + (o^_ ^o)) + ((o^_ ^o) + (o^_ ^o)) + ( ' ° ° ) [' ° ° ] + ((o^_ ^o) + (o^_ ^o)) + (o^_ ^o) + ( ' ° ° ) [' ° ° ] +
((o^_ ^o) + (o^_ ^o)) + ((o^_ ^o) + (o^_ ^o)) + ( ' ° ° ) [' ° ° ] + ( ' ° ° ) + ( ' - ) + ((o^_ ^o) + (o^_ ^o)) + ( ' ° ° ) [' ° ° ] +
((o^_ ^o) + (o^_ ^o)) + ((o^_ ^o) + (o^_ ^o)) + ( ' ° ° ) [' ° ° ] + ((o^_ ^o) + (o^_ ^o)) + ( ' - ) + ( ' ° ° ) [' ° ° ] + ((o^_ ^o)
+ (o^_ ^o)) + ((o^_ ^o) + (o^_ ^o)) + ( ' ° ° ) [' ° ° ] + ((o^_ ^o) + (o^_ ^o)) + ( ' - ) + ( ' ° ° ) + ( ' ° ° ) [' ° ° ] + ((o^_ ^o)
+ (o^_ ^o)) + ((o^_ ^o) + (o^_ ^o)) + ( ' ° ° ) [' ° ° ] + ( ' - ) + (o^_ ^o) + ( ' ° ° ) + ( ' ° ° ) [' ° ° ] + ((o^_ ^o) + (o^_ ^o))
+ ( ' - ) + (o^_ ^o) + ( ' ° ° ) [' ° ° ] + ((o^_ ^o) + (o^_ ^o)) + ( ' - ) + ( ' ° ° ) [' ° ° ] + ((o^_ ^o) + (o^_ ^o)) + ((o^_ ^o)
- ( ' ° ° )) + ( ' ° ° ) [' ° ° ] + ( ' ° ° ) + ( ' - ) + ( ' - ) + ( ' ° ° ) + ( ' ° ° ) [' ° ° ] + ((o^_ ^o) + (o^_ ^o)) + ((o^_ ^o) +
(o^_ ^o)) + ( ' ° ° ) [' ° ° ] + ( ' - ) + (o^_ ^o) + (c^_ ^o) + ( ' ° ° ) [' ° ° ] + ((o^_ ^o) + (o^_ ^o)) + ( ' - ) + (o^_ ^o)) +
( ' ° ° ) [' ° ° ] + ((o^_ ^o) + (o^_ ^o)) + ( ' - ) + ( ' ° ° ) [' ° ° ] + ((o^_ ^o) + (o^_ ^o)) + ((o^_ ^o) + (o^_ ^o)) + ( ' ° ° ) ['
° ° ] + ( ' ° ° ) + ( ' - ) + ( ' - ) + ( ' ° ° ) [' ° ° ] + ((o^_ ^o) + (o^_ ^o)) + ((o^_ ^o) + (o^_ ^o)) + ( ' ° ° ) [' ° ° ] + ( ' ° ° ) +
( ' - ) + (o^_ ^o) + ( ' ° ° ) [' ° ° ] ( ' ° ° ) ( ' ° ° ) ( ' ° ° )
```

直接在发布也找到提示了：



@tradzero 彩蛋可能是“第一行代码”，我是类推出来的结果



Kbyte 13 小时 4 分钟前

@byron tool.zcmzcm.org/aaencode



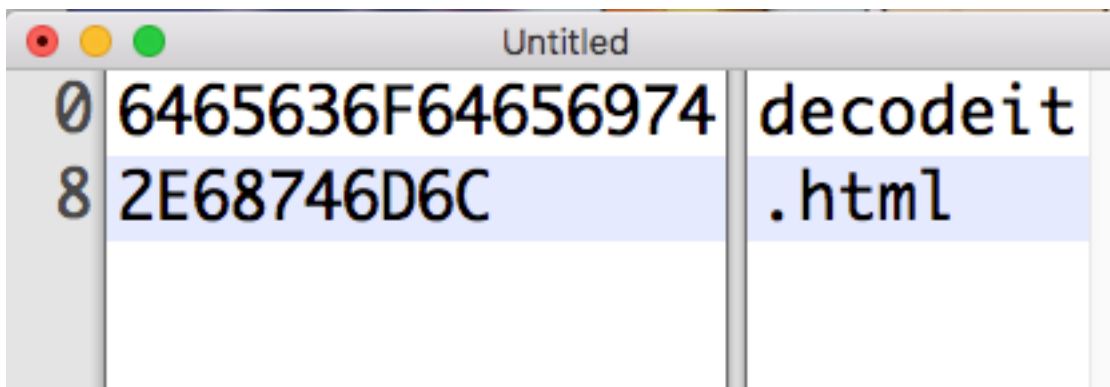
hhfans 12 小时 24 分钟前

aaencode 解密_aadecode

aaencode 加密 | 使用手册

6465636f646569742e68746d6c

再扔进16进制查看器：



Stage26:

<https://harrynull.tech/cipher/decodeit.html>

提示:

第二十六关 - 解码

<!--提示: 三重加密-->

<!--dial pad-->

MjM3MzQzODE3MTMyNDI5MjMyODE=

百度MjM3MzQzODE3MTMyNDI5MjMyODE= 出现:



Base64转换:

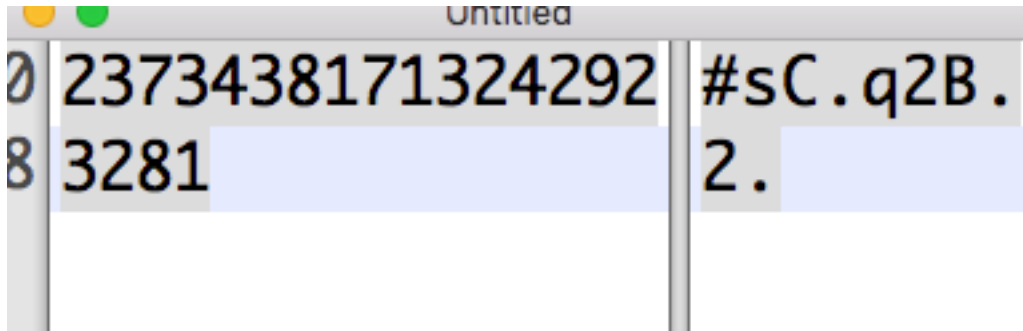


然而23734381713242923281并不是密码。

#sCq2B2也不是

ι 翎煖糝 ⊖ 同上

翎煖糝 同上



把数字转换为a~i的字母是：

b c g c d c h a g a c b d b i b h a

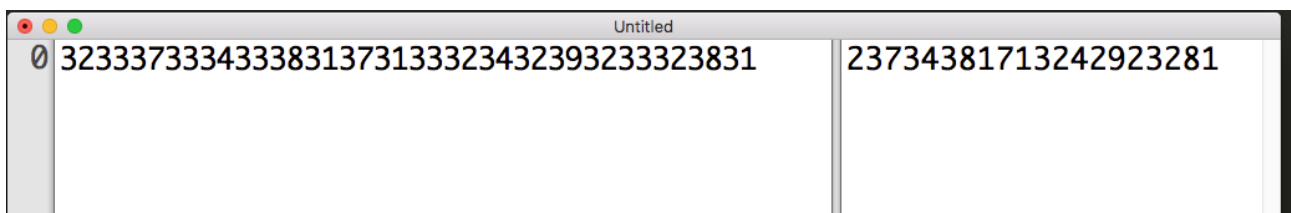
并不是密码

按照[列][行]的形式输入键盘上的字母是：

X M V I U D F L D I

x m v i u d f l d i

也不是密码



反向转换进制结果是

3233373334333831373133323432393233323831

```
>>> print(baseN(23734381713242923281, 30))
```

```
1ejobfdjnscldl
```

也不是密码

如果是凯撒密码的话，列出XMVIUDFLDI的所有可能：

PENAMVXDVA

QFOBNWYEWB

RGPCOXZFXC
SHQDPYAGYD
TIREQZBHZE
UJSFRACIAF
VKTGSBDJBG
WLUHTCEKCH
XMVIUDFLDI
YNWJVEGMEJ
ZOXKWFHNFK
APYLXGIOGL
BQZMYHJPHM
CRANZIKQIN
DSBOAJLRJO
ETCPBKMSKP
FUDQCLNTLQ
GVERDMOUMR
HWFSENPNVS
IXGTFOQWOT
JYHUGPRXPU
KZIVHQSYQV
LAJWIRTZRW
MBKXJSUASX
NCLYKTVBTY
ODMZLUWCUZ
PENAMVXDVA

都尝试了一遍似乎都不对。