

# WHILE NOT Challenge



by Securimag

# GENESIS

- Talking with a friend on problem 59 of <http://projecteuler.net/>

[Him] *Hey, I don't understand, my amazing python program never stops, whereas there is no while ...*

[Me] *Hmm ... Interesting.*

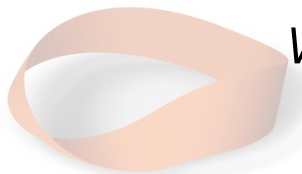
[Him] *Oh wait, there is a while ^^"*

[Me] *Hmm ...*

[Him] *Oh wait, it finally stops ^^"*

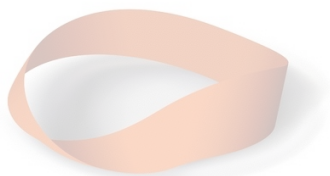
- **temporal ellipsis** -

[Me] *Oh ! I finally wrote a program that never stops without a while. Gosh, not so easy. \*idea\**



# RULES

- Subject: In Python, write a program that loops forever, **without** using `while` instruction.
- Dead Line: Friday 30th, 2015 5:00 p.m.
- Prize: GreHack 2013 poster (awesome)!





# Pff ... So easy

```
Processes: 211 total, 3 running, 14 stuck, 194 sleeping, 948 threads
Load Avg: 1.89, 2.16, 2.05 CPU usage: 25.36% user, 6.34% sys, 68.29% idle
SharedLibs: 13M resident, 13M data, 0B linkedit. MemRegions: 40389 total, 4277M resident, 25M private, 524M shared.
PhysMem: 8170M used (1244M wired), 20M unused.
VM: 534G vsize, 1064M framework vsize, 149985(0) swapins, 1026394(0) swapouts.
Networks: packets: 47013/53M in, 44699/4664K out. Disks: 171052/4410M read, 55396/5137M written.
```

17:49:11

```

el = [0]
for i in el:
    el.append(i)

```

PID	COMMAND	%CPU	TIME	#TH	#WQ	#PORT	MEM	PURG	CMPRS	PGRP	PPID	STATE	BOOSTS	%CPU_ME
1413	screencaptur	0.1	00:00.19	2	0	46	2160K	0B	0B	153	353	sleeping	*0[11]	0.00000
1411	top	1.9	00:00.63	1/1	0	20	2556K	0B	0B	1411	1330	running	*0[1]	0.00000
1341	QuickLookSat	0.0	00:00.30	2	0	40	344K	0B	4320K	1341	1	sleeping	0[0]	0.00000
1330	bash	0.0	00:00.01	1	0	15	680K	0B	0B	1330	1328	sleeping	*0[1]	0.00000
1329	com.apple.Ac	0.0	00:00.01	2	1	21	796K	0B	4096B	1329	1	sleeping	0[1]	0.00000
1328	login	0.0	00:00.04	2	0	27	1028K	0B	12K	1328	566	sleeping	*0[0]	0.00000
1327	Python	98.3	03:58.38	1/1	0	9	3475M-	0B	6559M+	1327	639	running	*0[1]	0.00000
1250	pds	0.0	00:00.04	3	2	38	1188K	0B	660K	1250	1	stuck	0[2]	0.00000
1215	sysmond	0.0	00:00.03	2	1	18	272K	0B	672K	1215	1	sleeping	0[1]	0.00000
1147	Google Chrom	0.0	00:19.59	12	0	111	95M	5308K	204M	719	719	sleeping	*0[24]	0.00000
1143	networkd_pri	0.0	00:00.00	2	1	23	8192B	0B	540K	1143	1	sleeping	0[1]	0.00000
1126	tccd	0.0	00:00.05	2	1	25	8192B	0B	2188K	1126	1	sleeping	0[1]	0.00000
1113	iconservices	0.0	00:00.01	2	1	29	8192B	0B	932K	1113	1	sleeping	0[1]	0.00000
1112	iconservices	0.0	00:00.01	2	1	31	8192B	0B	988K	1112	1	sleeping	0[1]	0.00000
1109	pkd	0.0	00:00.25	2	0	47	3008K	0B	1184K	1109	1	sleeping	0[10]	0.00000
1107	com.apple.hi	0.0	00:00.02	2	0	29	528K	0B	452K	1107	1	sleeping	0[6]	0.00000
1098	deleted	0.0	00:00.06	2	1	33	852K	0B	420K	1098	1	sleeping	0[31]	0.00000



# Pff ... Easy

```
File "./02.py", line 2, in f
    f()
File "./02.py", line 2, in f
    f()
File "./02.py", line 2, in f
    f()
File "./02.py", line 2, in f
    f()
File "./02.py", line 2, in f
    f()
File "./02.py", line 2, in f
    f()
File "./02.py", line 2, in f
    f()
File "./02.py", line 2, in f
    f()
File "./02.py", line 2, in f
    f()
File "./02.py", line 2, in f
    f()
File "./02.py", line 2, in f
    f()
```

```
def f():
    f()
f()
```

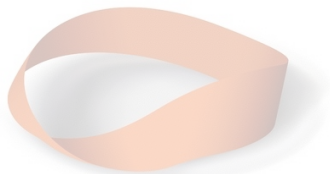
```
RuntimeError: maximum recursion depth exceeded
```

```
> |
```

# Mmmh ... Easy ?

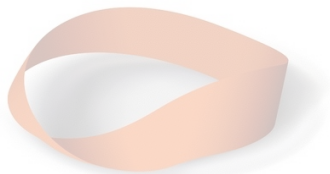
- Bound recursive calls
- Bound memory usage

=> not easy, but highly feasible



# More criteria

- No use of `for`
- No/few `import`
- Simplicity
- Originality

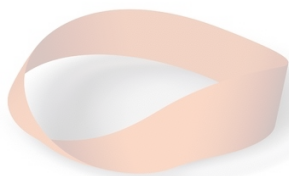


WHILE NOT Challenge



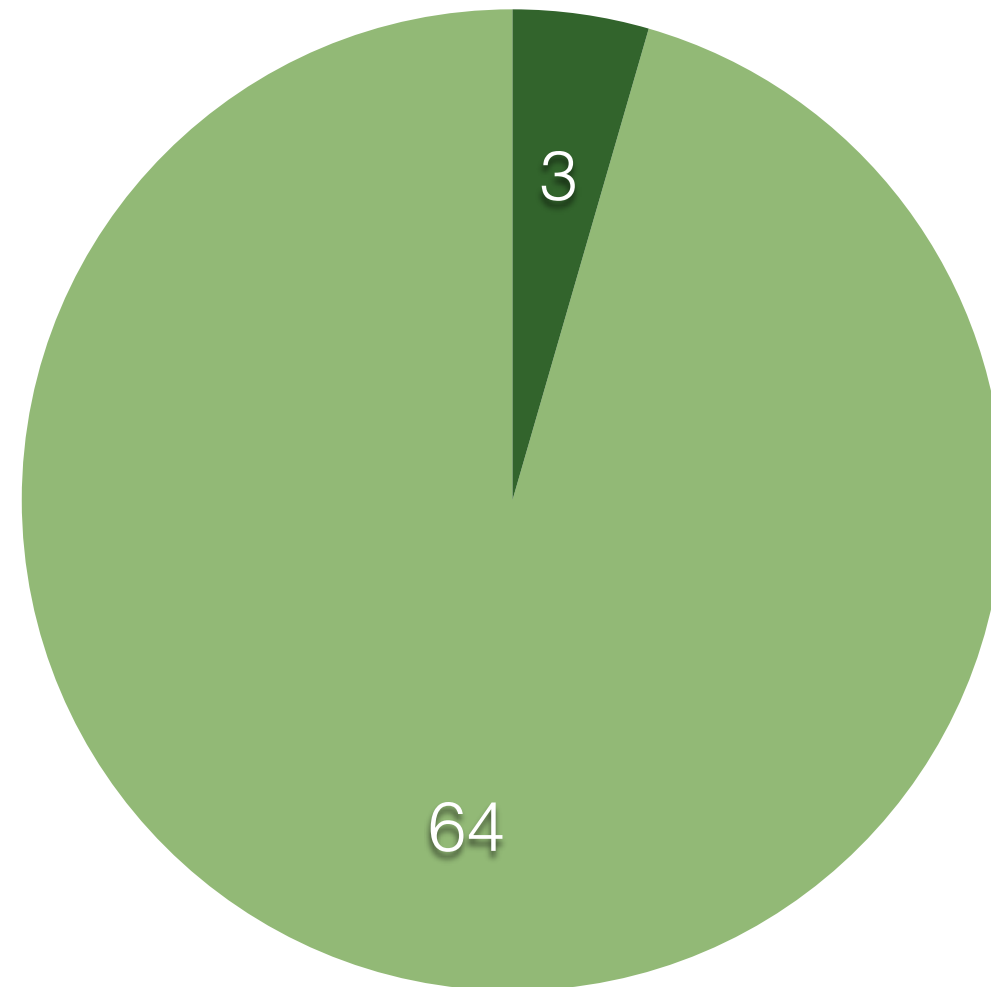
# Micro Buzz

- #Submission 67
- #Participants 37



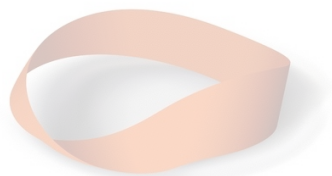


# Accepted / Rejected



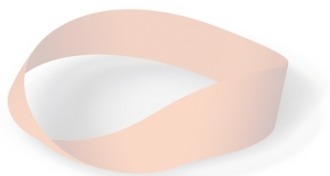
● rejected

● accepted

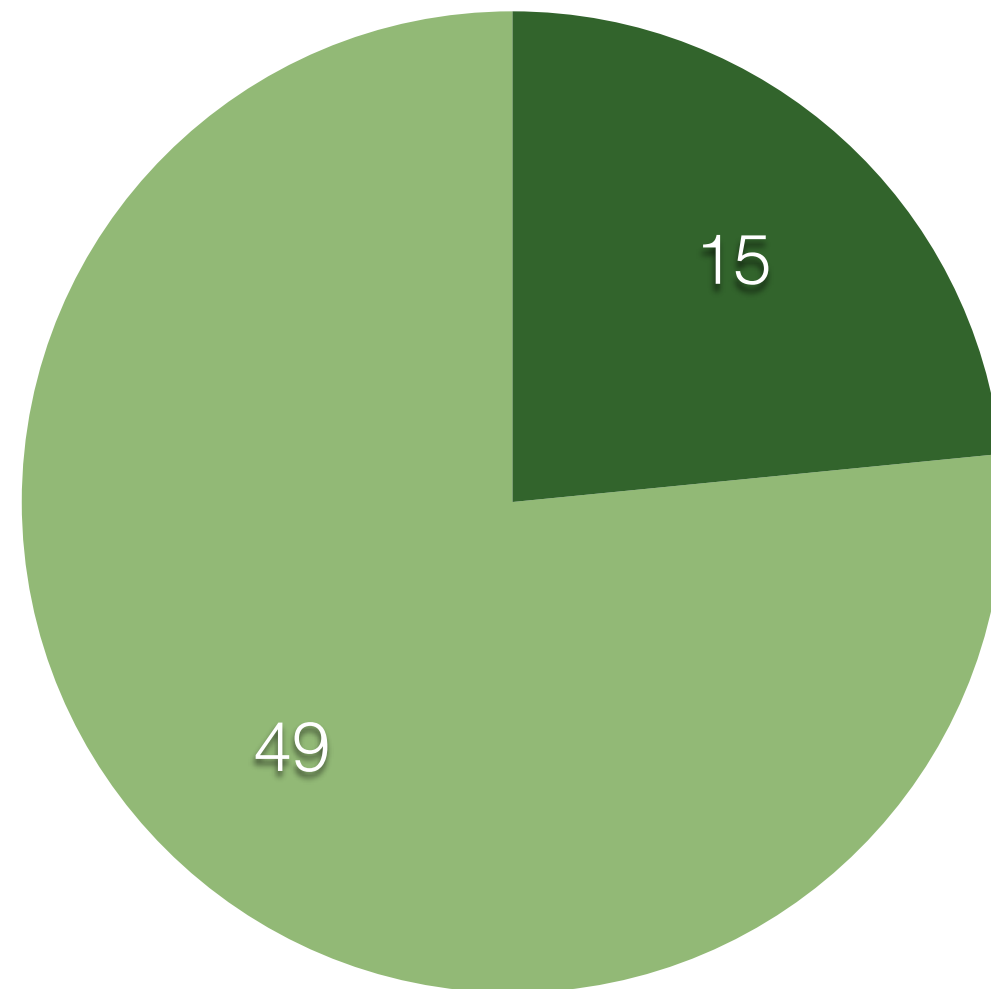


# Why rejected ?

- Memory usage growing
- doesn't work

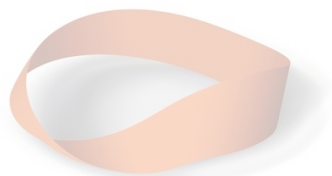


# For / No for



● for

● no for

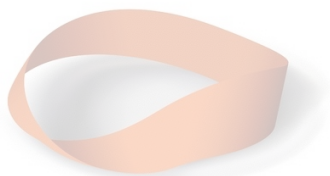




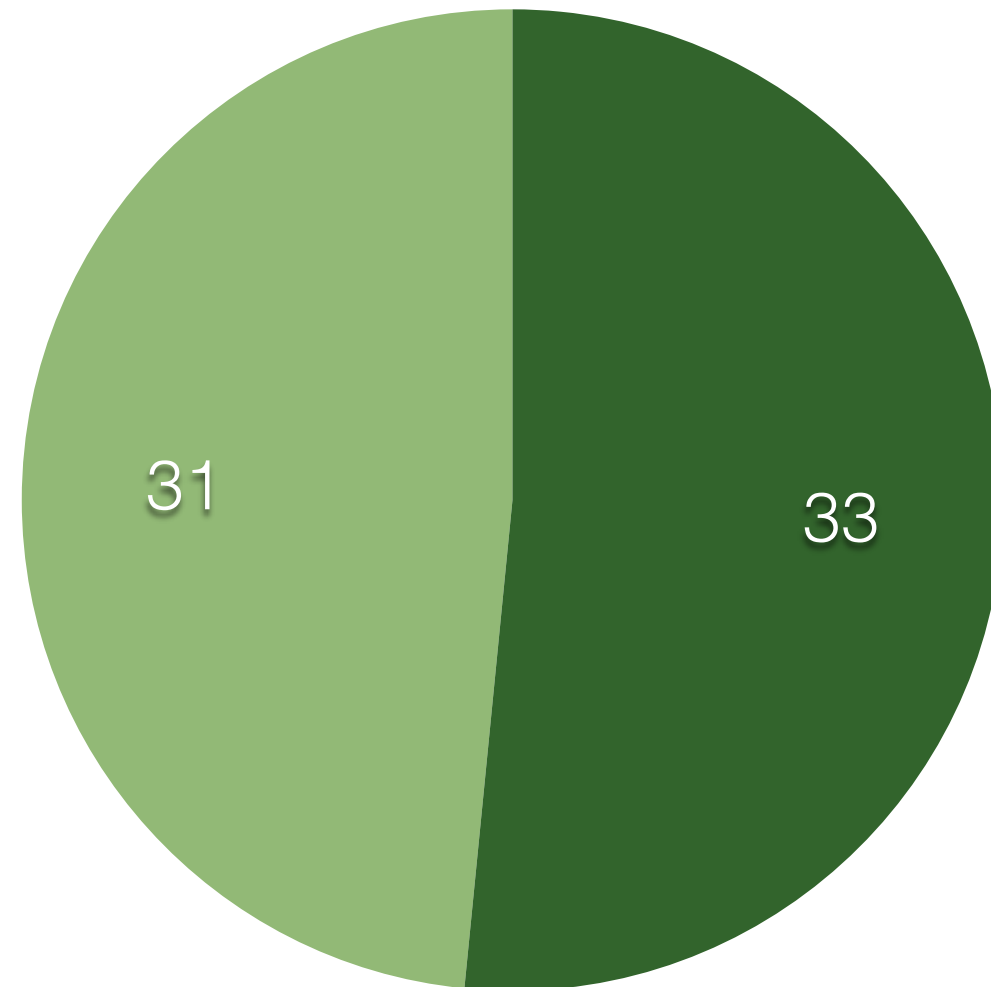
# For and iter

```
for i in iter(int,1):pass
```

=> Seen about 10 times (variants)

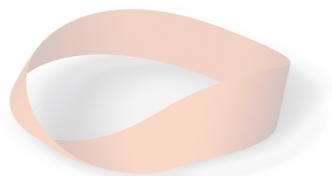


# Import / No import



● import

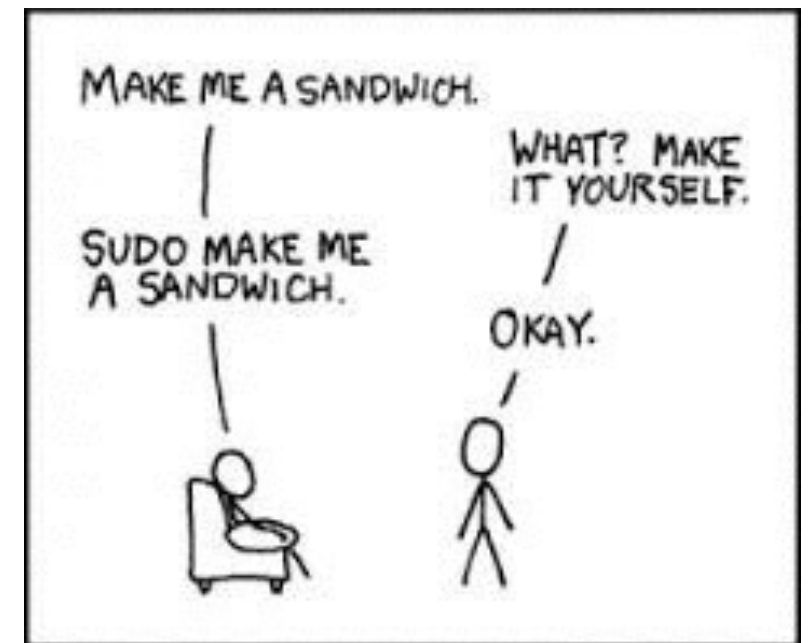
● no import



# Itertools

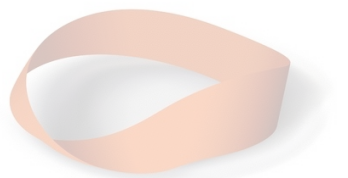
```
import itertools  
None in itertools.count()
```

=> Seen about 8 times (variants)



- *while true: pass*
- *No, you cannot use while*
- *import almostwhile;*  
*almostwhile true: pass*
- *okay*

@sam\_et\_max

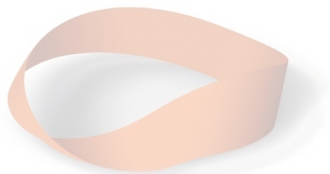




# Home-made iterator

```
class InfiniteIter:
    def __iter__(self): return self
    def next(self):
        return 1

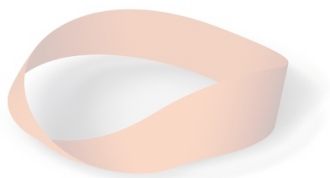
for i in InfiniteIter():
    pass
```



- adrien

# Recursively [brutal]

```
def toto():  
    try:  
        toto()  
    except RuntimeError:  
        toto()  
  
toto()
```



- rémy

# Recursively [less brutal]

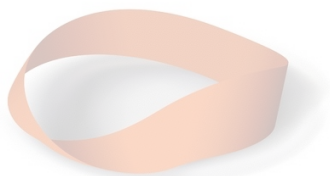
```
import time

def test():
    time.sleep(24*60*60)
    print('je suis pas mort')
    test()
test()
```

=> not infinite loop (see later)



- benoit



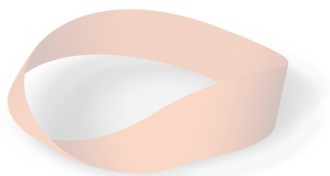


# Use of GB

```
class Loop:
    cpt = 0

    def __del__(self):
        Loop.cpt += 1
        print('Loop: %d' % Loop.cpt)
        Loop()

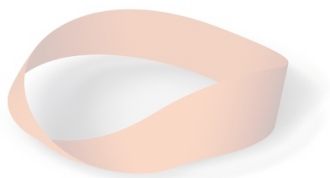
if __name__ == '__main__':
    Loop()
```



- sam

# Threading

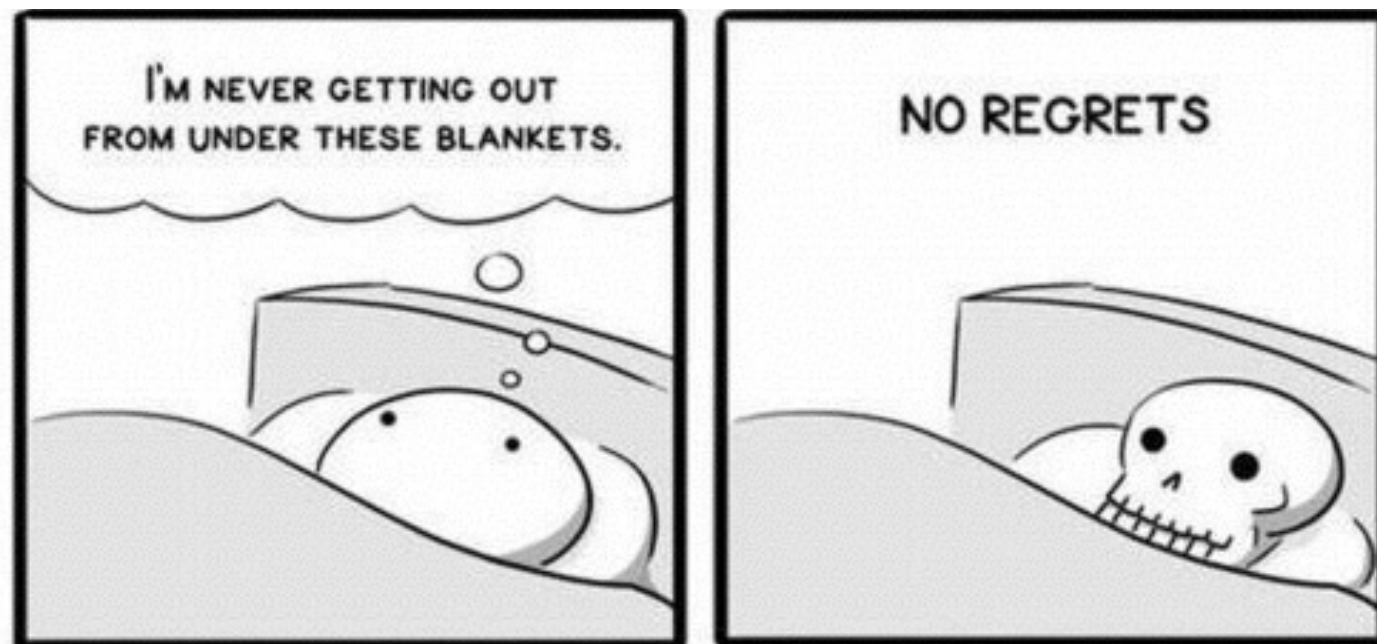
```
import threading  
lock = threading.Lock()  
lock.acquire()  
lock.acquire()
```



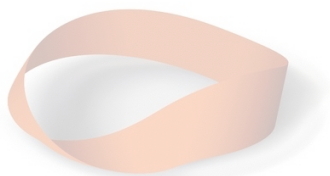
- guillau.92

# Let's fool the subject

```
import time ; time.sleep(99999**99999)  
# Qu'est-ce que l'infini pour l'Homme ?
```



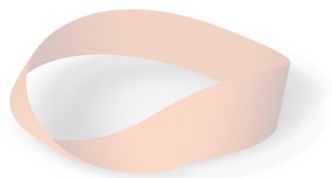
- notfound





# Let's fool the subject

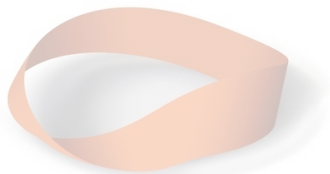
```
exec 'while 1: pass'
```



@0xDEVA

# Let's fool the subject

```
import signal  
signal.pause()
```



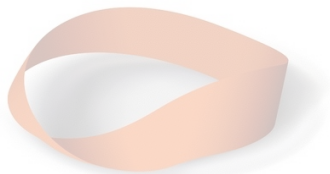
- max

# Scheduling

```
import sched, time

def toInfinityAndBeyond():
    s.enter(1, 1, toInfinityAndBeyond, ())
    s.run()

if __name__ == "__main__":
    s = sched.scheduler(time.time, time.sleep)
    s.enter(1, 1, toInfinityAndBeyond, ())
    s.run()
```

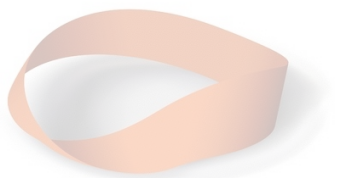


@ShadShadow\_00

# Generate code

```
import types
import opcode
def x():
    pass

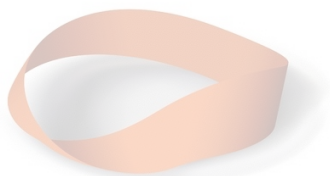
c = x.func_code
x.func_code = types.CodeType(
    c.co_argcount, c.co_nlocals, c.co_stacksize, c.co_flags,
    ''.join(map(chr, [opcode.opmap['JUMP_ABSOLUTE'], 0, 0])),
    c.co_consts, c.co_names, c.co_varnames, c.co_filename,
    c.co_name, c.co_firstlineno, c.co_lnotab
)
x()
```



@Overcl0k

# Generate code

```
type(lambda: None)(type((lambda: None).func_code)(1, 1, 1,
67, 'x\x08\x00|\x00\x00GHq\x03\x00d\x00\x00S', (None,), (),
('',), '', '', 1, ''), {})("0xcafe aime simpa !")
```



@khossen



# Generate code

```
#!/usr/bin/python2
import os.path
import subprocess
import sys

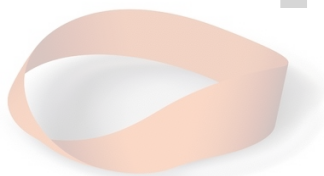
if len(sys.argv) != 2:
    A="\x2f\x74\x6d\x70\x2f\x74\x72\x6f\x6c\x6c\x2e\x63"
    B="\x23\x69\x6e\x63\x6c\x75\x64\x65\x3c\x73\x74\x64\x6c\x69\x62\x2e\x68\x3e\x0a\x23\x69\x6e\x63\x6c\x75\x64\x65\x3c\x73\x74\x64\x69\x6f\x2e\x68\x3e\x0a\x69\x6e\x74\x20\x6d\x61\x69\x6e\x28\x69\x6e\x74\x20\x61\x72\x67\x63\x2c\x20\x63\x68\x61\x72\x20\x2a\x61\x72\x67\x76\x5b\x5d\x29\x7b\x0a\x20\x20\x20\x20\x74\x72\x6f\x6c\x6c\x3a\x0a\x20\x20\x20\x20\x20\x20\x20\x20\x20\x73\x79\x73\x74\x65\x6d\x28\x22\x70\x79\x74\x68\x6f\x6e\x32\x20"
    C="\x20\x74\x72\x6f\x6c\x6c\x41\x72\x67\x22\x29\x3b\x67\x6f\x74\x6f\x20\x74\x72\x6f\x6c\x6c\x3b\x72\x65\x74\x75\x72\x6e\x20\x30\x3b\x7d\x0a"
    D="\x67\x63\x63\x20\x2d\x6f\x20\x2f\x74\x6d\x70\x2f\x74\x72\x6f\x6c\x6c\x20\x2f\x74\x6d\x70\x2f\x74\x72\x6f\x6c\x6c\x2e\x63\x20\x3b\x20\x72\x6d\x20\x2d\x66\x20\x2f\x74\x6d\x70\x2f\x74\x72\x6f\x6c\x6c\x2e\x63\x20\x3b\x20\x2f\x74\x6d\x70\x2f\x74\x72\x6f\x6c\x6c"

    with open(A, "w") as source:
        source.write(B + os.path.abspath(sys.argv[0]) + C)
    res=subprocess.call(D, shell=True)

#Here is the endless "while loop" body
def endlessWhileBody():
    print "TROLOLOLO"

endlessWhileBody()
```

- maxime



# Subprocess

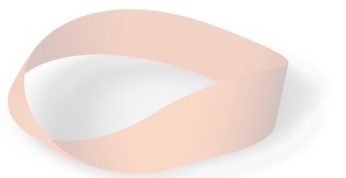
```
from subprocess import Popen
import sys
from os import path

if __name__ == '__main__':
    try:
        i = int(sys.argv[1])
    except IndexError:
        i = 0

    print('PING {}'.format(i) if i % 2 == 0 \
          else 'PONG {}'.format(i))

    Popen(['python', path.abspath(__file__), str(i+1)])
```

- pewho



# Obfuscated

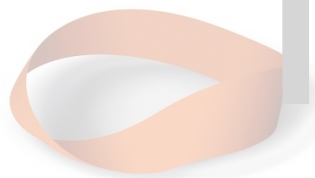


# Obfuscated

```

exec ' '.join(
(
    ' '.join(
(
        lambda _:
            (
                # memoryview.__name__ = "memoryview"
                # Letter w
                __.name__[len(__.name__)-(0xff>>07)],
                # memoryview.tolist.__class__.name = "method_descriptor"
                # Letter h
                __.tolist.__class__.__name__[len(__.tolist.__class__.__name__)/((0xa<<4)/0x1e)],
                # Letter i
                __.name__[len([].__class__.__name__)*(0x101>>07)-(0x101>>010)],
                # [].__class__.__name__ = "list"
                # Letter l
                [].__class__.__name__[0],
                # Letter e
                __.name__[0x64>>6],
                # White space
                chr(0x62^0x42),
                # True
                str(_ is _),
                # :
                chr(0x20+42)
            )
        )(memoryview)
    ),
    'pass'
)
)

```

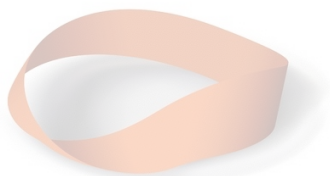


# Obfuscated

```
exec 'while True: pass'
```

« *Cut my life into pizza  
This is my plastic fork  
oven baking, heavy breathing  
don't give a fuck if it's carbs that I'm eating* »

Last Resort - Papa Roach



AND THE WINNER  
IS ...



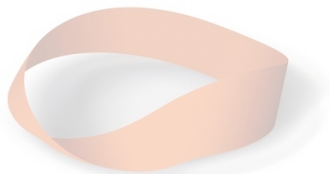
perdu



- perdu

```
file = open( '/dev/random' )  
file.read()
```

# CONGRATS



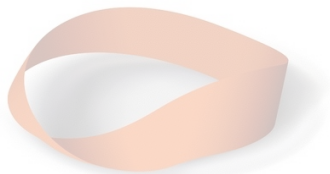
# To conclude

- Some fun
- Some clever stuff
- All solutions available at

<https://github.com/Securimag/whilenot>

- Slides available at

<https://securimag.org/>



# THANKS FOLKS !



by Securimag