```
class I:
  def __init__(self):
    self.s = 'abc'
    self.i = 0

  def __iter__(self):
    return self

  def __next__(self):
    if self.i == len(self.s):
      raise StopIteration
    v = self.s[self.i]
    self.i += 1
    return v


for x in I():
    print(x, end='')
```

```
abc
```

```
print(chr(ord('p') + 2))
```

```
r
```

```
class A:
    A = 1
    def __init__(self):
        self.a = 0


print(hasattr(A, 'a'))
```

```
False
```

```
assert var != 0
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-20-286d41719cbd> in <module>()
----> 1 assert var != 0
      2

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

✓ 0s completed at 4:54 PM

```
yes
no
```

```
from datetime import datetime

datetime_1 = datetime(2019, 11, 27, 11, 27, 22)
datetime_2 = datetime(2019, 11, 27, 0, 0, 0)

print(datetime_1 - datetime_2)


```

```
11:27:22
```

```
class A:
    def a(self):
        print('a')


class B:
    def a(self):
        print('b')


class C(B, A):
    def c(self):
        self.a()


o = C()
o.c()


```

```
b
```

```
import random

a = random.randint(0, 100)
b = random.randrange(10, 100, 3)
c = random.choice((0, 100, 3))
print(a, b, c)

a = random.choice((0, 100, 3))
b = random.randrange(10, 100, 3)
```

22

```
88 94 100
3 94 59
34 3 91
67 35 100
```

```
1  x = "\\\"
2  print(len(x))
```

```
  File "<ipython-input-30-328547a2b896>", line 1
    x = "\\\"
              ^
SyntaxError: EOL while scanning string literal
```

SEARCH STACK OVERFLOW

```
1  numbers = [i*i for i in range(5)]
2  # Insert line of code here.
3  foo = list(filter(lambda x: x % 2, numbers))
4  print(foo)
5
6  foo = list(map(lambda x: x % 2, numbers))
7  print(foo)
8
```

```
[1, 9]
[0, 1, 0, 1, 0]
```

```
1  %%file a.py
2  print("a", end='')
```

```
Writing a.py
```

```
1  %%file b.py
2  import a
3  print("b", end='')
```

```
Overwriting b.py
```

```
    def __init__(self, v=2):
        self.v = v

    def set(self, v=1):
        self.v += v
        return self.v


a = A()
b = a
b.set()
print(a.v)
```

```
3
```

```
try:
    raise Exception
except BaseException:
    print("a")
except Exception:
    print("b")
except:
    print("c")
```

```
a
```

```
import os

os.mkdir('pictures')
os.chdir('pictures')
```