

## RE: Simple Recursive Generator Question

**Source:** <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2003-12/2654.html>

---

**From:** Robert Brewer ([fumanchu\\_at\\_amor.org](mailto:fumanchu_at_amor.org))

**Date:** 12/19/03

Date: Fri, 19 Dec 2003 11:50:27 -0800

To: <[python-list@python.org](mailto:python-list@python.org)>

Probably not what you meant, but one of the things you're missing is that generators nullify the need for recursion in this case:

```
def big(mask):
    index = 0
    while True:
        if mask == 0:
            break
        if mask & 0x1:
            yield index
        index += 1
        mask = mask >> 1
```

```
>>> for x in big(0x16):
...     print x
...
1
2
4
```

The generator saves you having to pass the "index" param recursively. You save both object memory size (only one 'index' and one 'mask' object and their bound names) and function-calling overhead (stack space).

Robert Brewer  
MIS  
Amor Ministries  
[fumanchu@amor.org](mailto:fumanchu@amor.org)

```
> -----Original Message-----
> From: MetalOne [mailto:jcb@iteris.com]
> Sent: Friday, December 19, 2003 11:14 AM
> To: python-list@python.org
> Subject: Simple Recursive Generator Question
>
>
> I am trying to write a generator function that yields the
```

comp.lang.python: RE: Simple Recursive Generator Question

> *index position*  
> *of each set bit in a mask.*  
> *e.g.*  
> *for x in bitIndexGenerator(0x16): #10110*  
> *print x*  
> *--> 1 2 4*  
>  
>  
> *This is what I have, but it does not work.*  
> *Changing yield to print, shows that the recursion works correctly.*  
>  
> *def bitIndexGenerator(mask, index=0):*  
> *if mask == 0: return*  
> *elif mask & 0x1: yield index*  
> *bitIndexGenerator(mask >> 1, index+1)*  
>  
> *What am I missing?*  
> --  
> <http://mail.python.org/mailman/listinfo/python-list>  
>