

Re: better way to write this function

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2007-11/msg02673.html>

- *From:* Peter Otten <peter@xxxxxx>
 - *Date:* Mon, 26 Nov 2007 09:19:21 +0100
-

Kelie wrote:

Hello,

This function does I what I want. But I'm wondering if there is an easier/better way. To be honest, I don't have a good understanding of what "pythonic" means yet.

```
def divide_list(lst, n):
    """Divide a list into a number of lists, each with n items. Extra
    items are
    ignored, if any."""
    cnt = len(lst) / n
    rv = [[None for i in range(n)] for i in range(cnt)]
    for i in range(cnt):
        for j in range(n):
            rv[i][j] = lst[i * n + j]
    return rv
```

You can use slicing:

```
def chunks(items, n):
    .... return [items[start:start+n] for n in range(0, len(items)-n+1, n)]
    ....

for i in range(1,10):

    .... print chunks(range(5), i)
    ....
    [[0], [1], [2], [3], [4]]
    [[0, 1], [2, 3]]
    [[0, 1, 2]]
    [[0, 1, 2, 3]]
    [[0, 1, 2, 3, 4]]
    []
```

Re: better way to write this function

```
[]  
[]  
[]
```

Or build a generator that works with arbitrary iterables:

```
from itertools import *  
def chunks(items, n):  
  
    ... items = iter(items)  
    ... while 1:  
    ... chunk = list(islice(items, n-1))  
    ... chunk.append(items.next())  
    ... yield chunk  
    ...  
  
    list(chunks(range(5), 2))
```

```
[[0, 1], [2, 3]]
```

Peter

.