



链滴

Python 的发展路线图: 从学徒到大师

作者: [zhuangyan](#)

原文链接: <https://ld246.com/article/1373437702475>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```
<p>&nbsp;</p>
<p>&nbsp;</p>
<div>
  <ol>
    <li>Discover&nbsp;<a href="http://en.wikipedia.org/wiki/List_comprehension#Python">list
omprehensions</a> </li>
    <li>Discover&nbsp;<a href="http://en.wikipedia.org/wiki/Python_syntax_and_semantics#Ge
erators">generators</a> </li>
    <li>Incorporate&nbsp;<a href="http://docs.python.org/library/functions.html">map, reduce
filter, iter, range, xrange</a>&nbsp;often into your code</li>
    <li>Discover&nbsp;<a href="http://wiki.python.org/moin/PythonDecorators">Decorators<
a> </li>
    <li>Write recursive functions, a lot</li>
    <li>Discover&nbsp;<a href="http://docs.python.org/library/itertools.html">itertools</a>&
bsp;and&nbsp;<a href="http://docs.python.org/library/functools.html">functools</a> </li>
    <li>Read&nbsp;<a href="http://rads.stackoverflow.com/amzn/click/0596514980">Real Wor
d Haskell</a>&nbsp;<a href="http://book.realworldhaskell.org/read/">read free online</a>
</li>
    <li>Rewrite all your old Python code with tons of higher order functions, recursion, and wha
not.</li>
    <li>Annoy your cubicle mates every time they present you with a Python class. Claim it coul
be &quot;better&quot; implemented as a dictionary plus some functions. Embrace functional
programming.</li>
    <li>Rediscover the&nbsp;<a href="http://en.wikipedia.org/wiki/Strategy_pattern#Python">
trategy</a>&nbsp;pattern and then&nbsp;<a href="http://rads.stackoverflow.com/amzn/clc
/0596007124">all those things</a>&nbsp;from imperative code you tried so hard to forget a
ter Haskell.</li>
    <li>Find a balance.</li>
  </ol>
</div>
<p>&nbsp;</p>
<ol>
  <li>掌握 list comprehensions</li>
  <li>掌握 generators</li>
  <li>代码中经常使用 map, reduce, filter, iter, range, xrange</li>
  <li>掌握 Decorators</li>
  <li>大量使用递归函数</li>
  <li>掌握 itertools 和 functools</li>
  <li>读书 —— Real World Haskell (评论中作者说不一定要局限于这本书, 可以阅读任何关于函数
编程的书籍, 比如SICP等) </li>
  <li>重写自己的老派Python代码, 使用高阶函数, 递归等</li>
  <li>当你同事给你看任何用Python做的类实现时, 烦死他。告诉他更好的办法时通过dictionary加
函数, 拥抱函数编程。</li>
  <li>重新掌握 Strategy 及其他模式</li>
  <li>找到平衡</li>
</ol>
<p>http://stackoverflow.com/questions/2573135/python-progression-path-from-apprentice-
o-guru</p>
```