

如何学习 Python

作者: [melon](#)

原文链接: <https://ld246.com/article/1452760456328>

来源网站: [链滴](#)

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

Python语言相对Java来说，最大的特点就是易用易学，基本上每个linux系统都有安装python，像Java一样还需要自己下JVM、安装、设置环境变量等。有了其他编程语言的基础再来学习Python实是非常简单的，因为Python语言是基于[ABC](https://en.wikipedia.org/wiki/ABC_%28programming_language%29)语言开发出来的，ABC语言是以教学为目的的语言，其宗旨是让程序变得容易阅读和理解，所以Python语言也很容易理解和学习。

Python学习介绍

关于Python的学习网上有很多介绍，我在这里介绍一下我的学习过程，包括书籍，网站和一些工具等。

书籍

关于Python的书我推荐[《Python核编程（第二版）》](http://book.douban.com/subject/3112503/)，这本书看起来很厚其实读起来非常浅显易懂，重点可以看看下面这几章。

```
<div class="highlight">
```

```
<table>
```

```
<tbody>
```

```
<tr>
```

```
<td class="gutter">
```

```
<pre class="line-numbers"><span class="line-number">1</span>
```

```
<span class="line-number">2</span>
```

```
<span class="line-number">3</span>
```

```
<span class="line-number">4</span>
```

```
<span class="line-number">5</span>
```

```
<span class="line-number">6</span>
```

```
<span class="line-number">7</span>
```

```
<span class="line-number">8</span>
```

```
<span class="line-number">9</span>
```

```
<span class="line-number">10</span>
```

```
<span class="line-number">11</span>
```

```
<span class="line-number">12</span>
```

```
</pre>
```

```
</td>
```

```
<td class="code">
```

```
<pre><code class="sh"><span class="line">第4章 Python对象
```

```
</span><span class="line">6.8 Unicode
```

```
</span><span class="line">8.11 迭代器和iter<span class="o">()</span>函数
```

```
</span><span class="line">第9章 文件的输入和输出
```

```
</span><span class="line">第10章 错误和异常
```

```
</span><span class="line">第11章 函数和函数式编程
```

```
</span><span class="line">第12章 模块
```

```
</span><span class="line">第13章 面向对象编程
```

```
</span><span class="line">第14章 执行环境
```

```
</span><span class="line">第15章 正则表达式
```

```
</span><span class="line">第18章 多线程编程
```

```
</span><span class="line">20.2 使用Python进行Web应用：创建一个简单的Web客户端
```

```
</span></code></pre>
```

```
</td>
```

```
</tr>
```

```
</tbody>
```

```
</table>
```

```
</div>
```

如果不想看这种太多理论的书的话，可以看这本实战的书[《Learn Python The Hard Way》](http://learnpythontheharday.org/)，里面的例子从浅到深让你一步步的学习Python比较适合没有编程经验的新手。里面的例子不要只是把它们打出来，最好自己可以举一反三做一些关

知识点的挑战和扩展。 </p>

<h4>网站</h4>

<p>除了看书编码外，平时业余时间可以订阅一些Python的blog和网站，下面是我最近收集的一些Python博客，分享一下:</p>

 Planet Python: 这是最出名的python博客其中之一，快去看看，我希望你能找到对你有用的东西。

 lucumr: 博主是flask（一个python web框架）的始人

 Doug Hellmann: 博主是PYMOTW(Python Module Of the Week)成员之一，博客里面包含了很多python library的知识。如果你是在寻找一些实用python库的话，你可以在这里找找。

 Mouse Vs Python: 这个博客更新也比较频繁。

 pydanny: 这是一个主要关注Django的博客，但有很多关于Python的东西。

<h4>工具 & 技巧</h4>

<p>pip和setuptools就不说了，Python开发必备，下载管理依赖包就靠他们了，其他最有用的Python工具要数virtualenv了，它可以创建一个虚拟的Python执行环境，因为Python的虚拟机不像JVM那么大，创建出来之后文件夹非常小（大概只有10M），在这个虚拟环境里面安装执行Python程序不会污染你操作系统的主python环境。 </p>

<div class="highlight">

<table>

<tbody>

<tr>

<td class="gutter">

<pre class="line-numbers">1

2

3

4

5

6

7

8

</pre>

</td>

<td class="code">

<pre><code class="sh"># 安装virtualenv

pip install virtualenv

创建Python虚拟环境

virtualenv venv

让虚拟环境生效

source venv/bin/activate

取消虚拟环境

deactivate

</code></pre>

</td>

</tr>

</tbody>

</table>

</div>

<p>另外Python的调试也非常方便，只要在需要设置断点的地方添加代码<code>import pdb;pdb.s

`t_trace()`即可，然后执行程序就会在断点处停止，你可以通过打印下面命令进行调试:

 `c`: 让程序继续往下走，continue的意思

 `s`: 进入子方法内部，相当于step into

 `n`: 跳到下一句 这里介绍的只是一些基本的命令，更多的命令可以看[这里](https://docs.python.org/2/library/pdb.html)，另外还可以打印断点处的各种变量和执行语句。

<h2>简洁的Python</h2>

<p>在Python中执行`import this`可以看到Python之禅。</p>

<blockquote>

<p>The Zen of Python, by Tim Peters</p>

<p>Beautiful is better than ugly.&br />Explicit is better than implicit.&br />Simple is better than complex.&br />Complex is better than complicated.&br />Flat is better than nested.&br />Sparse is better than dense.&br />Readability counts.&br />Special cases aren't special enough to break the rules.&br />Although practicality beats purity.&br />Errors should never pass silently.&br />Unless explicitly silenced.&br />In the face of ambiguity, refuse the temptation to guess.&br />There should be one—and preferably only one—obvious way to do it.&br />Although that way may not be obvious at first unless you're Dutch.&br />Now is better than never.&br />Although never is often better than *right* now.&br />If the implementation is hard to explain, it's a bad idea.&br />If the implementation is easy to explain, it may be a good idea.&br />Namespaces are one honking great idea — let's do more of those!</p>

</blockquote>

<p>从Python之禅中我们可以看到Python是一门追求简洁，追求优雅的语言，语言设计者对代码质量要求比较高，希望编写Python代码的开发人员也能写出高质量的代码。</p>

<p>由于Python简洁而强大的函数式编程，我们可以很方便的创建结构稍微复杂的对象，特别复杂还是建议使用class来创建对象。</p>

<p>比如我们有这样的一个数据结构，一个字典里面包含多个集合，用Java来初始化大概是这个样子:</p>

<div class="highlight">

<table>

<tbody>

<tr>

<td class="gutter">

<pre class="line-numbers">1

2

3

4

5

6

7

8

9

10

</pre>

</td>

<td class="code">

<pre><code class="java">Map<String,List<String>>map=nwHashMap<String,List<&

```

; </span> <span class="n">String</span> <span class="o">&gt;&gt;(); </span>
</span> <span class="line"> <span class="n">List</span> <span class="o">&lt;</span> <span
n class="n">String</span> <span class="o">&gt;</span> <span class="n">list1</span> <s
an class="o">=</span> <span class="k">new</span> <span class="n">ArrayList</span><
pan class="o">&lt;</span> <span class="n">String</span> <span class="o">&gt;&lt;(); </span>
</span> <span class="line"> <span class="n">list1</span> <span class="o">.</span> <span
lass="na">add</span> <span class="o">(</span> <span class="s">"foo1"</span> <span cla
s="o">); </span>
</span> <span class="line"> <span class="n">list1</span> <span class="o">.</span> <span
lass="na">add</span> <span class="o">(</span> <span class="s">"bar1"</span> <span cla
s="o">); </span>
</span> <span class="line"> <span class="n">map</span> <span class="o">.</span> <span
class="na">put</span> <span class="o">(</span> <span class="s">"list1"</span> <span cla
s="o">,</span> <span class="n">list1</span> <span class="o">); </span>
</span> <span class="line"> <span class="n">List</span> <span class="o">&lt;</span> <span
n class="n">String</span> <span class="o">&gt;</span> <span class="n">list2</span> <s
an class="o">=</span> <span class="k">new</span> <span class="n">ArrayList</span><
pan class="o">&lt;</span> <span class="n">String</span> <span class="o">&gt;&lt;(); </span>
</span> <span class="line"> <span class="n">list2</span> <span class="o">.</span> <span
lass="na">add</span> <span class="o">(</span> <span class="s">"foo2"</span> <span cla
s="o">); </span>
</span> <span class="line"> <span class="n">list2</span> <span class="o">.</span> <span
lass="na">add</span> <span class="o">(</span> <span class="s">"bar2"</span> <span cla
s="o">); </span>
</span> <span class="line"> <span class="n">map</span> <span class="o">.</span> <span
class="na">put</span> <span class="o">(</span> <span class="s">"list2"</span> <span cla
s="o">,</span> <span class="n">list2</span> <span class="o">); </span>
</span> </code> </pre>
</td>
</tr>
</tbody>
</table>
</div>
<p>而用Python一行代码就可以搞定:</p>
<div class="highlight">
<table>
<tbody>
<tr>
<td class="gutter">
<pre class="line-numbers"> <span class="line-number">1</span>
</pre>
</td>
<td class="code">
<pre> <code class="python"> <span class="n">my_map</span> <span class="o">=</span>
<span class="p">{</span></span> <span class="s">'list1'</span> <span class="p">:</span> <span
lass="p">[</span> <span class="n">foo1</span> <span class="p">,</span> <span class="n">
bar1</span> <span class="p">],</span> <span class="s">'list2'</span> <span class="p">:
/</span> <span class="p">[</span> <span class="s">'foo2'</span> <span class="err">,</sp
n> <span class="s">'bar2'</span> <span class="p">]}</span>
</code> </pre>
</td>
</tr>
</tbody>
</table>

```

</div>

<p>所以说Java是一门很啰嗦的语言，虽然新版本的Java加了lambda，然而并不能减少多少Java语本身的繁杂性。</p>

<h2>Djaogo学习介绍</h2>

<p>Python来实现一些小工具小应用是十分方便的，但Python也可以用来做Web开发，Python比有名的Web框架有 Django, Flask, Tornado等，其中Djaongo使用最为广泛，集成的东西也比较，不管你是使用关系数据库还是非关系数据库，是否使用缓存等都可以使用Django，其最好的一特点是集成了管理员功能，可以省却开发者很大一部分开发量。</p>

<p>有人可能会说Django比较重，对于新手来说学习成本比较高，对于这个我没有什么意见，因为作需要使用Django，对于其他的框架还没有接触过，但我使用后的体验是Django虽然要配置的东西较多，但只要配置好了，开发效率还是挺快的，也可能是我以前大部分时间使用Spring来开发，相对pring来说Django的配置还算比较轻的，所以对我来说没有什么感觉:)</p>

<p>学习Django可以上Django的官网，上面有startup的向导，通过向导一步的操作可以让你快速了解一个web项目的开发，另外上面还有很多文档，可以知道Django的所有容。</p>

<p>另外一个学习Django比较好的是《Django Book》这本书，这里有它的网站，书很浅显易读，详细讲解了Django各个部分的内，有时候是先通过一个比较简单的实现方案实现需求，然后利用Django的特性让代码得到优化，更的代码来实现更多的内容，这也体现了Django的思想，简洁优雅，复用性高。</p>

<p>理论知识学习的差不多了，最后当然是做项目练手了，可以自己动手做一个web项目来熟悉Python和Django，在实际操作的过程中遇到不懂的问题就google一下，然后记录下来，做过一两个项目后你应该对Python和Django比较了解了:)</p>