# 在线资源的优先级

虽然有很多关于编程的在线资源，但是并非所有资源都是同等水平的。下面的资源列表按照大致的可靠性顺序排序。

1. [Python 教程](https://docs.python.org/3/tutorial/" \t "https://classroom.udacity.com/nanodegrees/nd002-cn-basic/parts/ce2f97bf-7be1-4142-a764-9b59ec87985f/modules/3c5137f9-0620-41c4-b575-283ab5f45482/lessons/bd8a41f3-dd73-4b89-b9fc-05a48638efbb/concepts/_blank) - 这部分官方文档给出了 Python 的语法和标准库。它会举例讲解，并且采用的语言比主要文档的要浅显易懂。确保阅读该文档的 Python 3 版本！
2. [Python 语言和库参考资料](https://docs.python.org/3/index.html" \t "https://classroom.udacity.com/nanodegrees/nd002-cn-basic/parts/ce2f97bf-7be1-4142-a764-9b59ec87985f/modules/3c5137f9-0620-41c4-b575-283ab5f45482/lessons/bd8a41f3-dd73-4b89-b9fc-05a48638efbb/concepts/_blank) - 语言参考资料和库参考资料比教程更具技术性，但肯定是可靠的信息来源。当你越来越熟悉 Python 时，应该更频繁地使用这些资源。
3. **第三方库文档** - 第三方库会在自己的网站上发布文档，通常发布于 [https://readthedocs.org/](https://readthedocs.org/" \t "https://classroom.udacity.com/nanodegrees/nd002-cn-basic/parts/ce2f97bf-7be1-4142-a764-9b59ec87985f/modules/3c5137f9-0620-41c4-b575-283ab5f45482/lessons/bd8a41f3-dd73-4b89-b9fc-05a48638efbb/concepts/_blank) 。你可以根据文档质量判断第三方库的质量。如果开发者没有时间编写好的文档，很可能也没时间完善库。
4. **非常专业的网站和博客** - 前面的资源都是主要资源，他们是编写相应代码的同一作者编写的文档。主要资源是最可靠的资源。次要资源也是非常宝贵的资源。次要资源比较麻烦的是需要判断资源的可信度。[Doug Hellmann](https://doughellmann.com/blog/" \t "https://classroom.udacity.com/nanodegrees/nd002-cn-basic/parts/ce2f97bf-7be1-4142-a764-9b59ec87985f/modules/3c5137f9-0620-41c4-b575-283ab5f45482/lessons/bd8a41f3-dd73-4b89-b9fc-05a48638efbb/concepts/_blank) 等作者和 [Eli Bendersky](http://eli.thegreenplace.net/" \t "https://classroom.udacity.com/nanodegrees/nd002-cn-basic/parts/ce2f97bf-7be1-4142-a764-9b59ec87985f/modules/3c5137f9-0620-41c4-b575-283ab5f45482/lessons/bd8a41f3-dd73-4b89-b9fc-05a48638efbb/concepts/_blank) 等开发者的网站很棒。不出名作者的博客可能很棒，也可能很糟糕。
5. [StackOverflow](http://stackoverflow.com/" \t "https://classroom.udacity.com/nanodegrees/nd002-cn-basic/parts/ce2f97bf-7be1-4142-a764-9b59ec87985f/modules/3c5137f9-0620-41c4-b575-283ab5f45482/lessons/bd8a41f3-dd73-4b89-b9fc-05a48638efbb/concepts/_blank) - 这个问答网站有很多用户访问，因此很有可能有人之前提过相关的问题，并且有人回答了！但是，答案是大家自愿提供的，质量参差不齐。在将解决方案应用到你的程序中之前，始终先理解解决方案。如果答案只有一行，没有解释，则值得怀疑。你可以在此网站上查找关于你的问题的更多信息，或发现替代性搜索字词。
6. **Bug 跟踪器** - 有时候，你可能会遇到非常罕见的问题或者非常新的问题，没有人在 StackOverflow 上提过。例如，你可能会在 GitHub 上的 bug 报告中找到关于你的错误的信息。这些 bug 报告很有用，但是你可能需要自己开展一些工程方面的研究，才能解决问题。
7. **随机网络论坛** - 有时候，搜索结果可能会生成一些自 2004 年左右就不再活跃的论坛。如果这些资源是唯一解决你的问题的资源，那么你应该重新思考下寻找解决方案的方式。