**下载文件**

作者: **莫烦**编辑: **莫烦 2017-12-30**

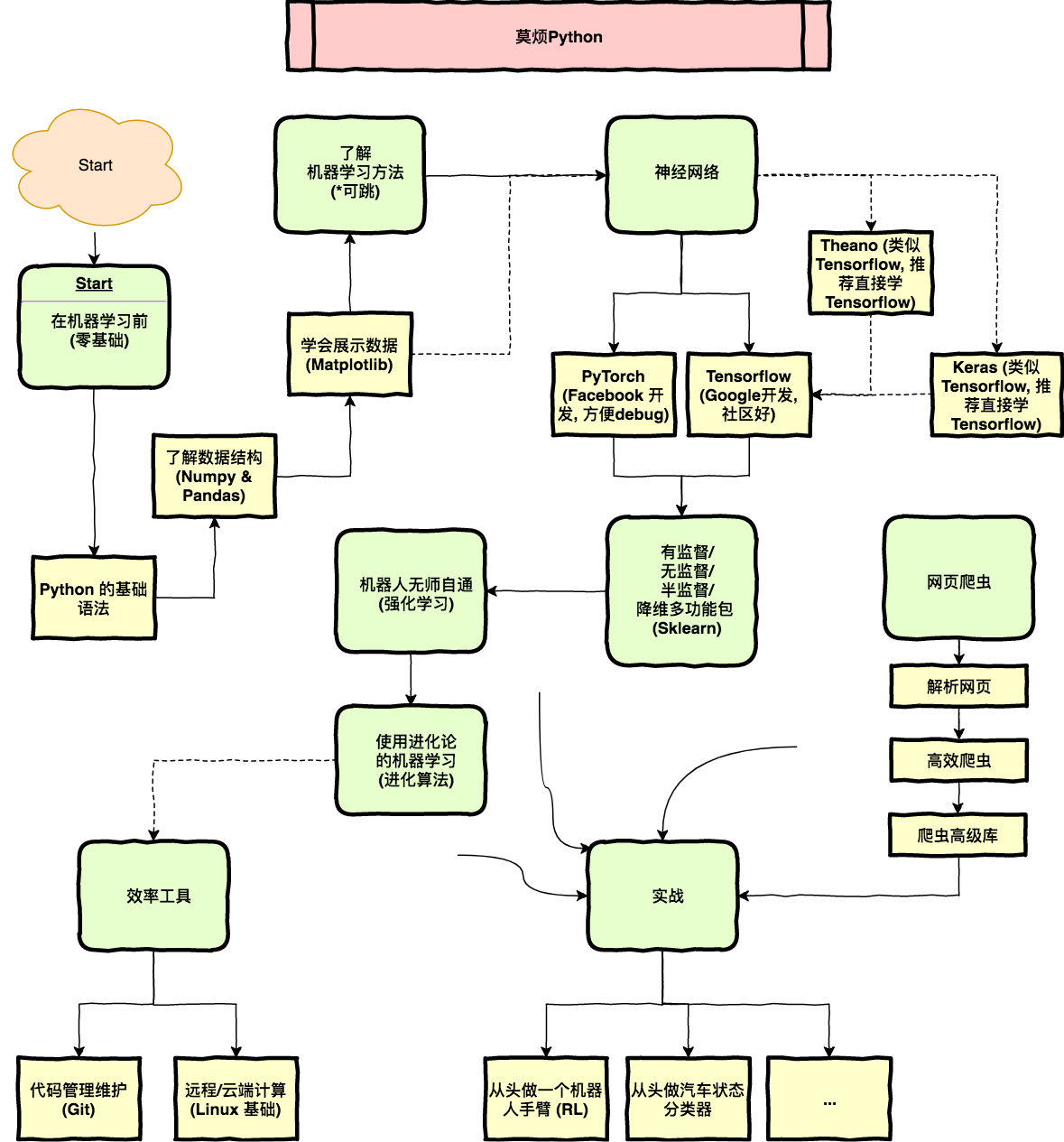
学习资料:

* [本节学习代码](https://github.com/MorvanZhou/easy-scraping-tutorial/blob/master/notebook/3-2-download.ipynb)

从网上下载文件的时候你肯定很开心, 比如说什么电影, pdf, 音乐等. 我们使用爬虫, python 一样也可以做到. 而且做得途径还有很多. 今天我们就来探讨有哪些可行的途径, 之后我们还能用这些途径来做一个小实战, [下载国家地理杂志上的美图](https://morvanzhou.github.io/tutorials/data-manipulation/scraping/3-03-practice-download-image/).

[**下载之前**](https://morvanzhou.github.io/tutorials/data-manipulation/scraping/3-02-download/#%E4%B8%8B%E8%BD%BD%E4%B9%8B%E5%89%8D)

在下载之前, 我们的弄清楚怎么样下载. 打个比方, 以前有朋友留言说我的教程太多, 不知道从何学起, 我专门制作了一张学习流程图, 这张[莫烦Python的个性化学习路线](https://morvanzhou.github.io/learning-steps/)应该也拯救了无数迷途少年吧. 哈哈哈, 今天我们来爬这张图, 还有下载这张图.

[](https://morvanzhou.github.io/static/img/description/learning_step_flowchart.png)

想下这张图, 我们首先要到这张图所在的[网页](https://morvanzhou.github.io/learning-steps/). 在这个网页中找到这张图的位置, 并右键 inspect, 找到它在 HTML 中的信息.

[](https://morvanzhou.github.io/static/results/scraping/3-2-1.png)

发现原图被存放在这个网页, 注意这个地址开头是 /, 并不是完整的网址, 这种形式代表着, 它是在 “https://morvanzhou.github.io/” 下面的网址. 所以我们还要将其补全, 才能在网址栏中找到这个图片地址.

src="/static/img/description/learning\_step\_flowchart.png"

补全后的[网址](https://morvanzhou.github.io/static/img/description/learning_step_flowchart.png)是:

https://morvanzhou.github.io/static/img/description/learning\_step\_flowchart.png

找到了这个网址, 我们就能开始下载了. 为了下载到一个特定的文件夹, 我们先建立一个文件夹吧. 并且规定这个图片下载地址.

**import** os

os.makedirs('./img/', exist\_ok=True)

IMAGE\_URL = "https://morvanzhou.github.io/static/img/description/learning\_step\_flowchart.png"

[**使用 urlretrieve**](https://morvanzhou.github.io/tutorials/data-manipulation/scraping/3-02-download/#%E4%BD%BF%E7%94%A8-urlretrieve)

在 urllib 模块中, 提供了我们一个下载功能 urlretrieve. 使用起来很简单. 输入下载地址 IMAGE\_URL 和要存放的位置. 图片就会被自动下载过去了.

**from** urllib.request **import** urlretrieve

urlretrieve(IMAGE\_URL, './img/image1.png')

[**使用 request**](https://morvanzhou.github.io/tutorials/data-manipulation/scraping/3-02-download/#%E4%BD%BF%E7%94%A8-request)

而在 [requests](https://morvanzhou.github.io/tutorials/data-manipulation/scraping/3-01-requests/) 模块, 也能拿来下东西. 下面的代码实现了和上面一样的功能, 但是稍微长了点. 但我们为什么要提到 requests 的下载呢? 因为使用它的另一种方法, 我们可以更加有效率的下载大文件.

**import** requests

r = requests.get(IMAGE\_URL)

with open('./img/image2.png', 'wb') as f:

f.write(r.content)

所以说, 如果你要下载的是大文件, 比如视频等. requests 能让你下一点, 保存一点, 而不是要全部下载完才能保存去另外的地方. 这就是一个 chunk 一个 chunk 的下载. 使用 r.iter\_content(chunk\_size) 来控制每个 chunk 的大小, 然后在文件中写入这个 chunk 大小的数据.

r = requests.get(IMAGE\_URL, stream=True) # stream loading

with open('./img/image3.png', 'wb') as f:

for chunk in r.iter\_content(chunk\_size=32):

f.write(chunk)

有了这些知识的积累, 我们就能开始做一个小的实战练习, 去[下载一些国家地理杂志的美图](https://morvanzhou.github.io/tutorials/data-manipulation/scraping/3-03-practice-download-image/)啦.