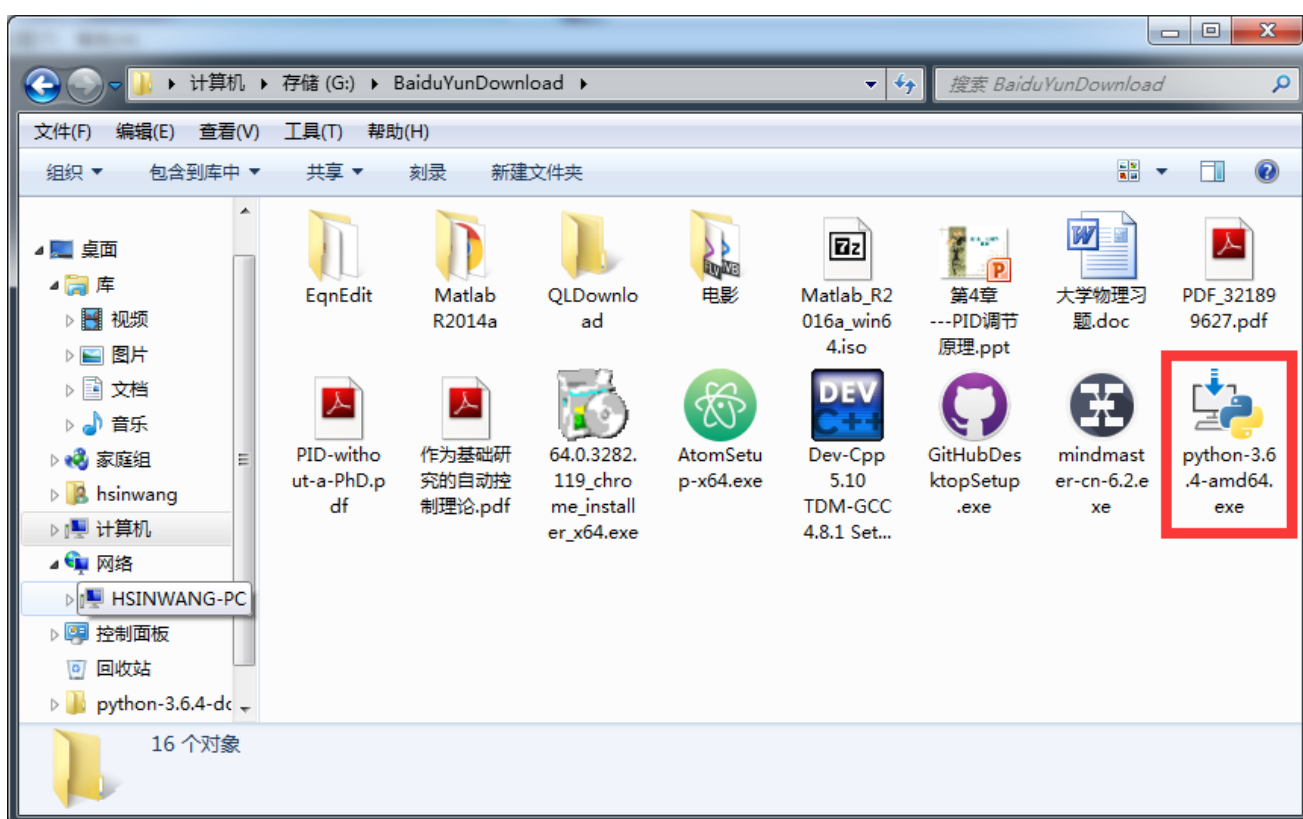


- 1 python的安装包获取和一般安装
- 2 pip的安装和配置
- 3 自动化全部包的更新

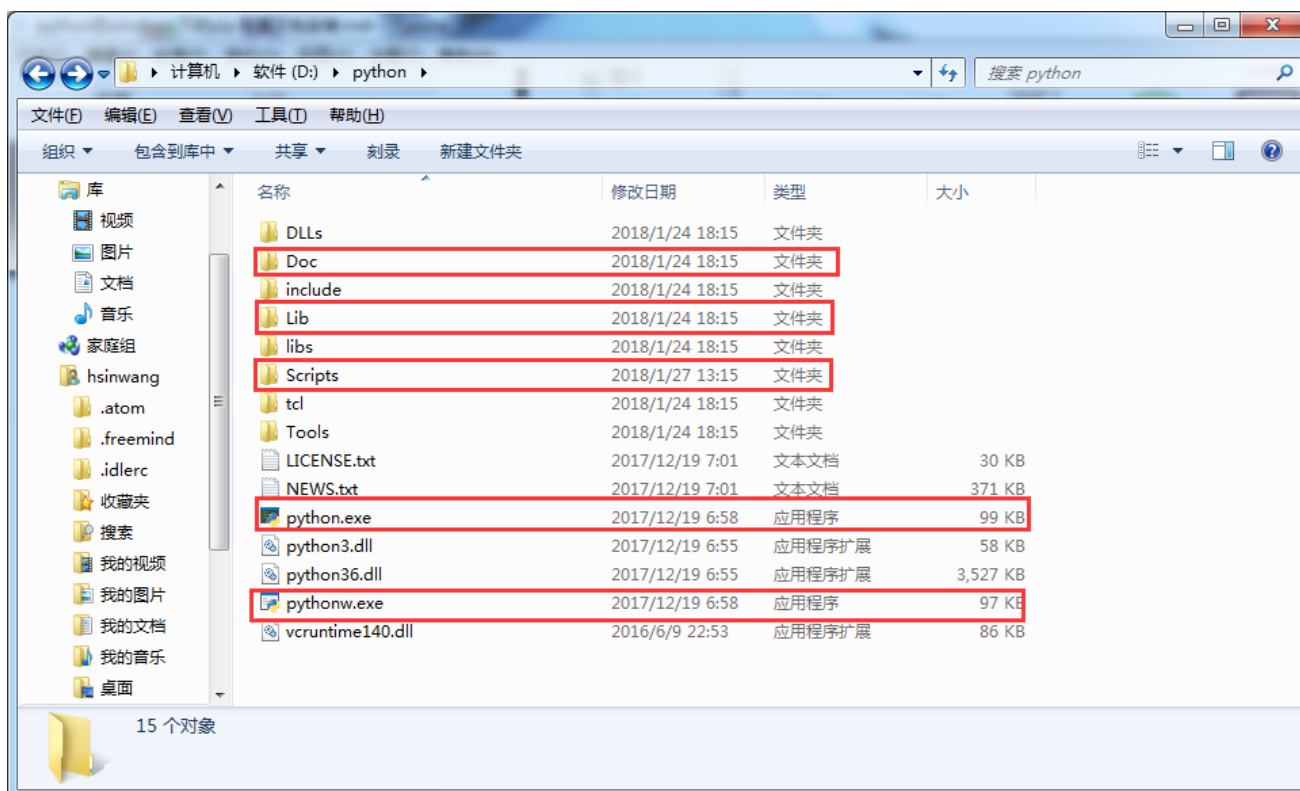
## 1 python的安装包获取和一般安装

python官网获取 64位或者32位安装包

[python.org](https://python.org)



我下载的是3.6.4, 3.x或者2.x差别不大, 但是如果机器性能差一些, 或者想和以前的老代码交互的话, 建议2.x; 但是现在3.x的包, 以及一些特别新的算法什么的封装多是3.x, 强行用2.x也没问题, 官方有2.x>>3.x的对照表和自动化函数修正包, (虽然没用过但是应该是可以用的). 安装软件没什么可说的, 但是有一点最好分配全体user, 否则后面需要管理员权限安装那些包就很麻烦了. 安装完成后应该是 (c/d/g--):xx\python\---- (任意盘符下的一个python文件夹). 如下。



## 2 pip的安装和配置

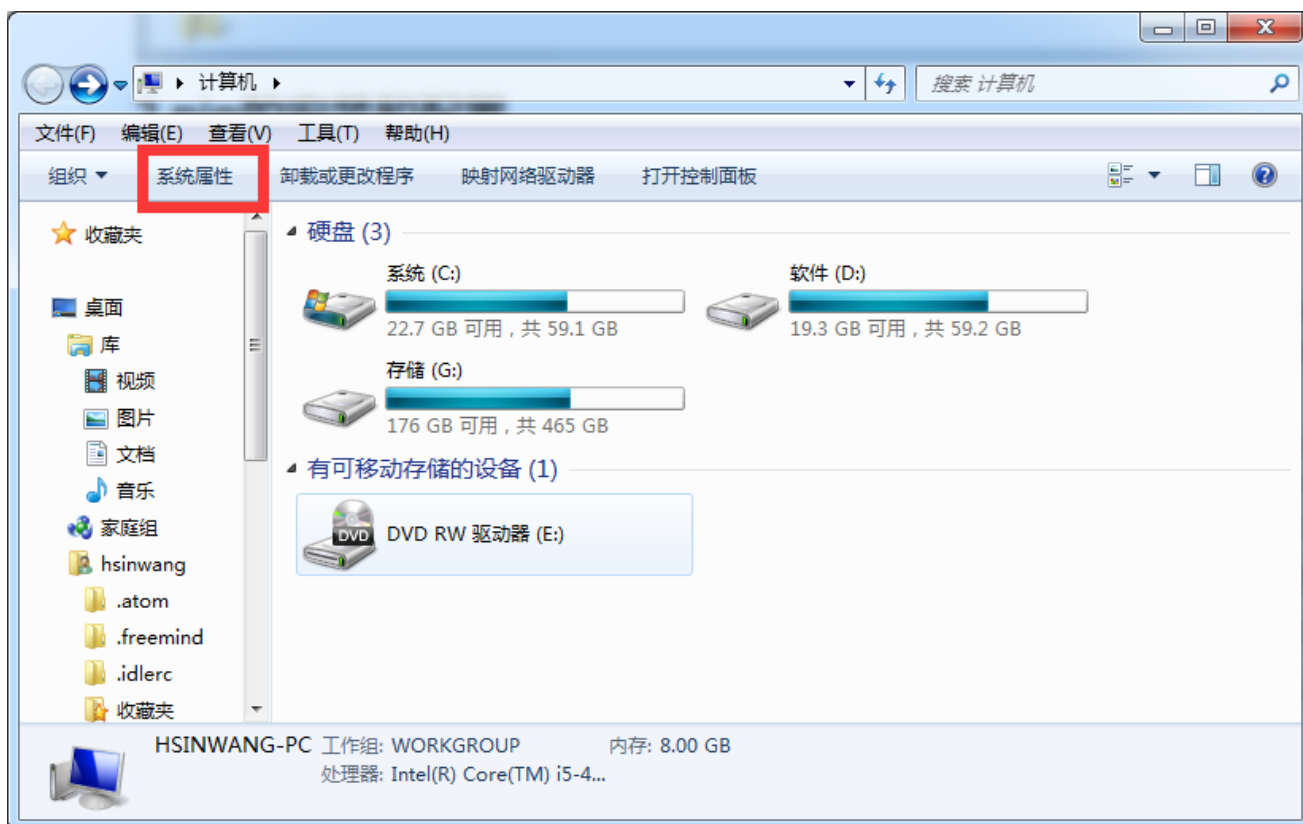
上图中的几个红色框框内的文件夹和可执行文件分别是以下作用

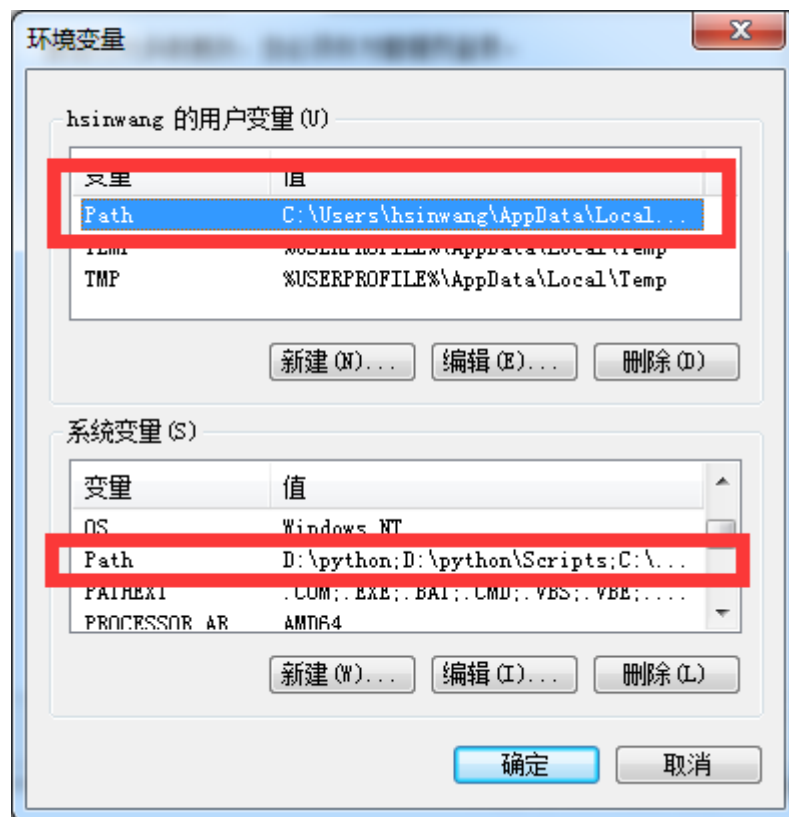
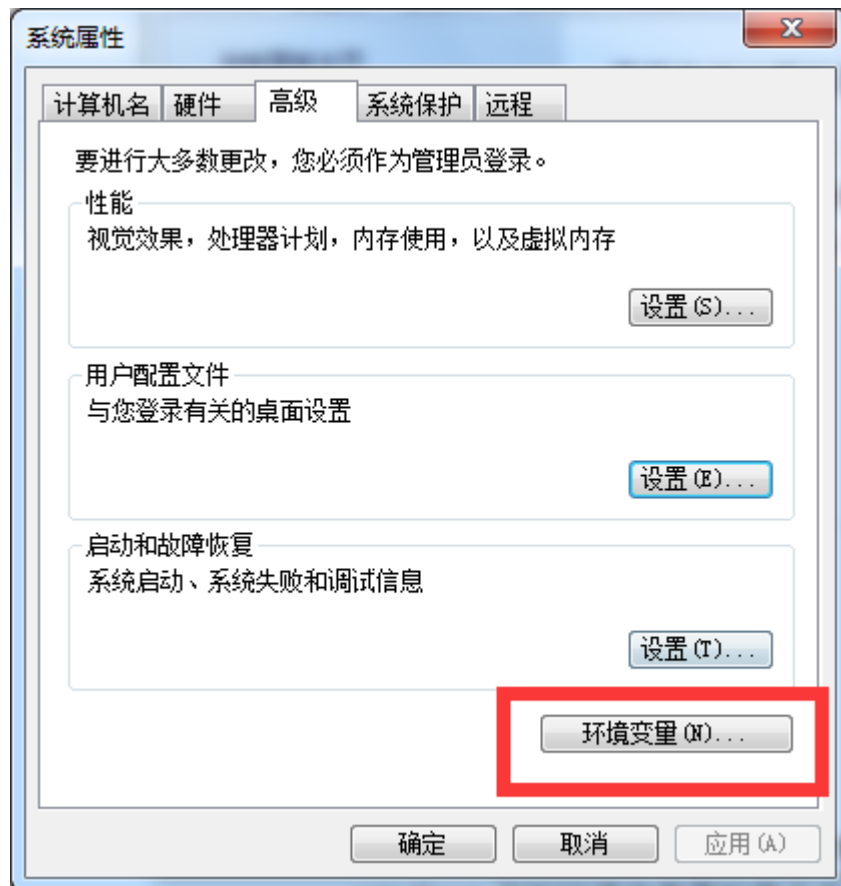
- Doc : python 的官方和社区共同编写的手册（函数，类，包的查询）
- Lib : 内置函数类型（内置定义 结构体 函数 数据结构 文件读写 常用数学运算），里面的有一个文件夹需要注意：site-packages, 这个文件存储的是官方和社区共同发行的包（发行包就是每个人都参与更改最终形成一定范围内使用的子程序集），pip 所下载的 程序包自动化的默认的安装路径就是这里。
- Scripts ; 脚本，就是可以直接执行的命令存储的文件 windows 下的命令主要是command命令和shell,像linux下的Bash etc--可以在window安装但是都需要 cygwin 这类模拟终端。用法和win的命令没什么区别，而且都自带man，只要愿意耐心的看下说明书，都可以明白。命令的集合就是脚本了。etc : Batch
- python.exe : 编码的即时验证，交互式命令行，主要是写函数时不确定的验证，查询对象，类，包的用法
- pythonw.exe : 同上，多了一个官方的文本编辑器IDE，一边调试一边写效率会低，但是写的东西bug少

pip 在3.x的python版本下是默认放在Scripts文件夹里了，而2.x就需要自己下载安装了，安装全都默认，它会自动找到你的python并且链接它的，完全不用手动配置。

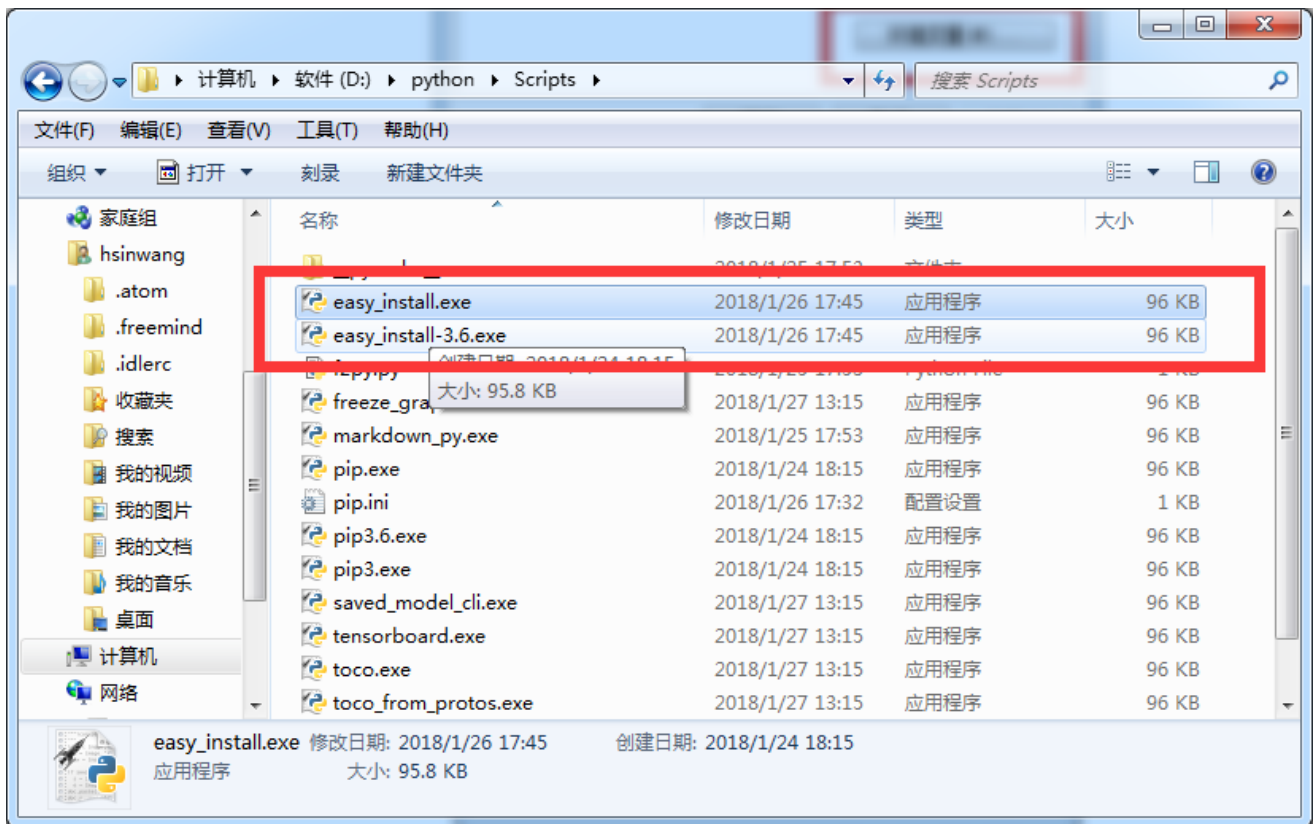
1 pip命令的绝对地址设置

windows 下的环境变量真的是无奈了，好多软件都是这样需要自己设定App-path.





最后一张图的用户变量和环境变量都加上 xx :\python ;xx:\python\Scripts; 一个是python的命令行，一个python 脚本的命令行，主要是使用pip;另外



这个easy\_install.exe 也是安装很多包必备的工具，一般如果大部分设置是默认的话，这个就是已经安装过了，不用担心。现在pip 安装已经完成了，但是使用的时候有时候会出现下载包不成功的时候，这是以为镜像在国外的原因，所以要改用国内的源。新建一个pip.ini 放在Scripts文件夹下,编辑如下：

```

1 [global]
2     ; 全局变量
3 timeout = 6000
4     ; 每60秒断开尝试重连
5 index-url = https://pypi.tuna.tsinghua.edu.cn/simple
6     ; 引用源的索引页面
7 trusted-host = https://pypi.tuna.tsinghua.edu.cn
8     ; 引用源的主机地址
9 ; 很多大学和互联网公司都提供映像，我的个人使用经验是 还是用大学的好一些，速度快而且稳定
10 ; 教育网的优势就在这里了，而且ipv6的DNS越来越多，速度和连接会越来越快，东软的源速度也快

```

配置完成就可以愉快的下载 包 了

win +R ,cmd ,熟悉的命令行 ：输入pip

```
C:\Windows\system32\cmd.exe
C:\Users\hsinwang>pip

Usage:
  pip <command> [options]

Commands:
  install          Install packages.
  download         Download packages.
  uninstall        Uninstall packages.
  freeze           Output installed packages in requirements format.
  list             List installed packages.
  show             Show information about installed packages.
  check            Verify installed packages have compatible dependen
cies.
  search           Search PyPI for packages.
  wheel            Build wheels from your requirements.
  hash             Compute hashes of package archives.
  completion       A helper command used for command completion.
  help             Show help for commands.

General Options:
  -h, --help       Show help.
  --isolated       Run pip in an isolated mode, ignoring
environment variables and user configuration.
  -v, --verbose    Give more output. Option is additive, and can be
used up to 3 times.
  -U, --version    Show version and exit.
  -q, --quiet      Give less output. Option is additive, and can be
used up to 3 times (corresponding to WARNING,
ERROR, and CRITICAL logging levels).
  --log <path>    Path to a verbose appending log.
  --proxy <proxy> Specify a proxy in the form
[user:passwd@]proxy.server:port.
  --retries <retries> Maximum number of retries each connection should
attempt (default 5 times).
  --timeout <sec> Set the socket timeout (default 15 seconds).
  --exists-action <action> Default action when a path already exists:
(s)witch, (i)gnore, (w)ipe, (b)ackup, (a)bort.
  --trusted-host <hostname> Mark this host as trusted, even though it does
not have valid or any HTTPS.
  --cert <path>   Path to alternate CA bundle.
  --client-cert <path> Path to SSL client certificate, a single file
containing the private key and the certificate
```

常用的就是红色框的三条命令 安装 列表 和帮助。这里有个问题。这些包是动态更新的，但是pip仅仅提供了单个包的安装 更新 和卸载。所以有必要自己写一个全部包的更新脚本。

### 3 自动化全部包的更新

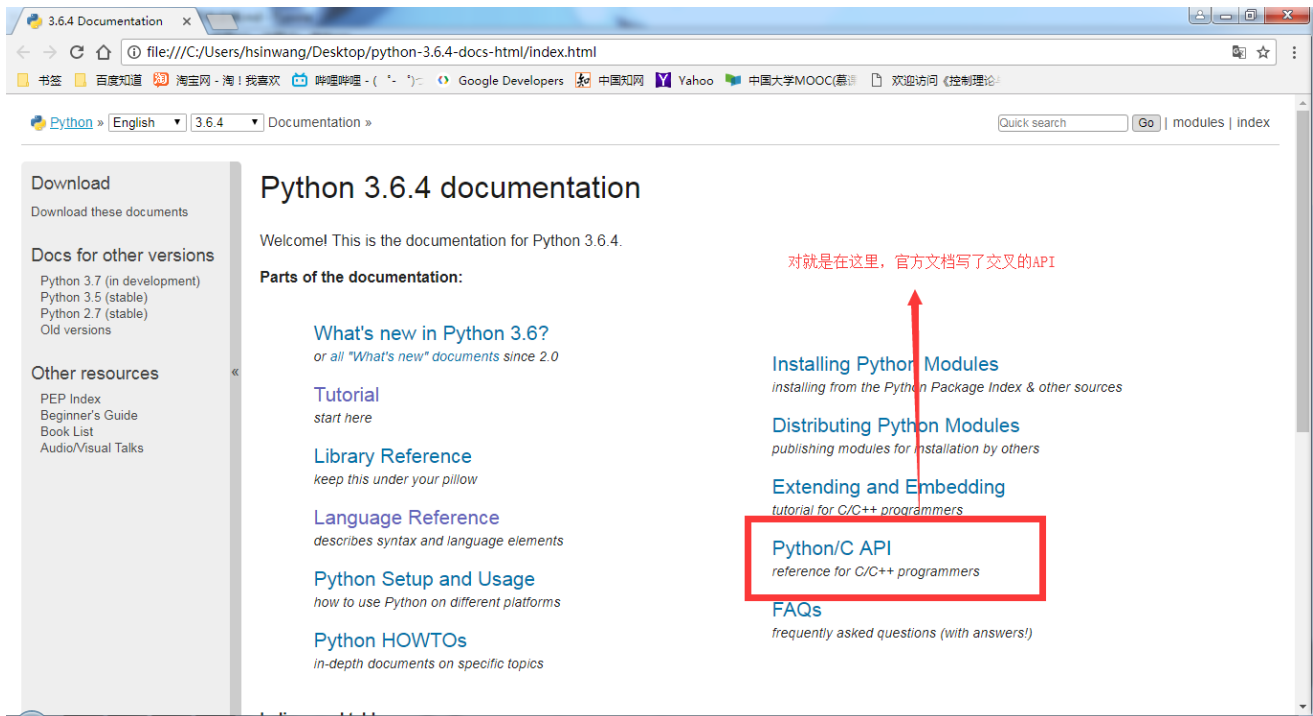
任意地方新建一个update.py,我把它也放在Scripts文件夹下，编写：

```

1 import pip
2 from subprocess import call
3
4 for dist in pip.get_installed_distributions():
5     call("pip install --upgrade " + dist.project_name, shell=True)
6     .....
7         遍历所有的包 , 每次都调用 pip命令
8     .....

```

Tip :配置就完了, 各种安装包有的是有依赖关系的,所以最好按照逻辑关系从上到下安装, 防止发生版本之间不兼容的状况, 写这个东西主要是windows 太烦人, 每次版本更新都要再来一次很嫌弃, 没有linux的更新机制那么容易, 脚本越多函数越来越复杂, 图形化会让事情变得复杂, 以上在win的种种配置问题, 用linux几个命令一个文件就全部解决了。搞了几天matlab,我发现它的有些机制不好理解, 要继续看, 弄懂它的状态函数机制。还有python的编译器 解释器等都是c编写出来的, 所以它也提供了c的很多东西来加快执行速度。开心的写扩展23333



hsinwang