# AutoCmd使用手册

## 安装python环境

### Window下安装：

Window，Python环境的安装请参考：<https://blog.csdn.net/weixin_40844416/article/details/80889165>

#### 1.1在python的官网下载python对应版本：

https://www.python.org/downloads/windows/

64位下载Windows x86-64 executable installer 版本

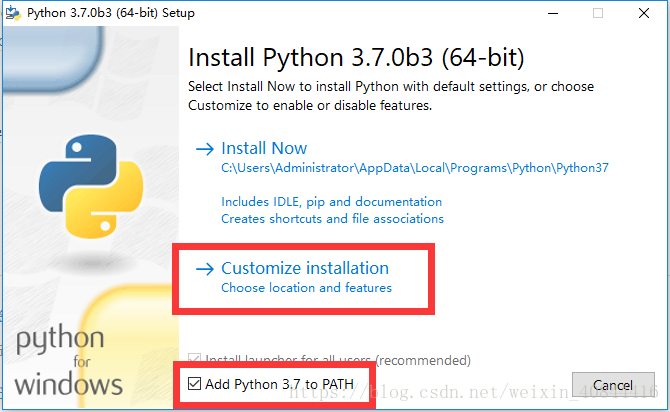
32位下载Windows x86 executable installer 版本

打开链接如下图，版本会一直更新，选择任意一个适合自己电脑的版本就好

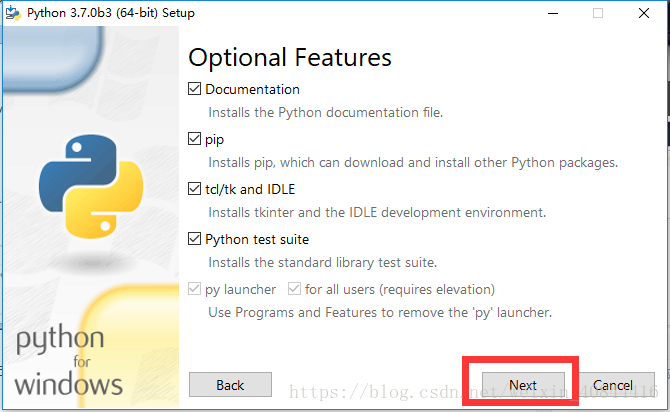


#### 1.2勾选 Add python to PATH 添加路径

安装界面点击Customize installation 自定义安装，记得勾选加入环境变量。

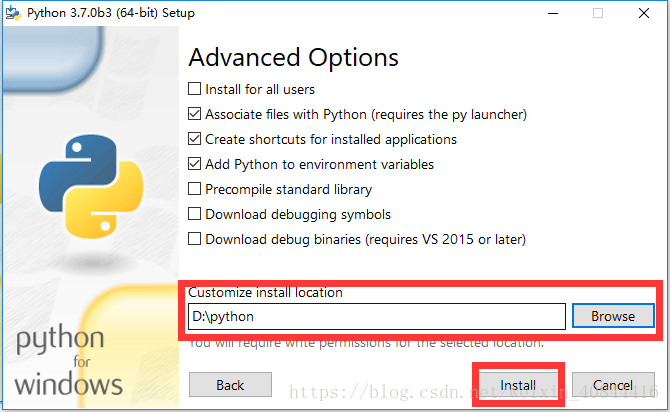


#### 1.3不改变默认进行Next下一步

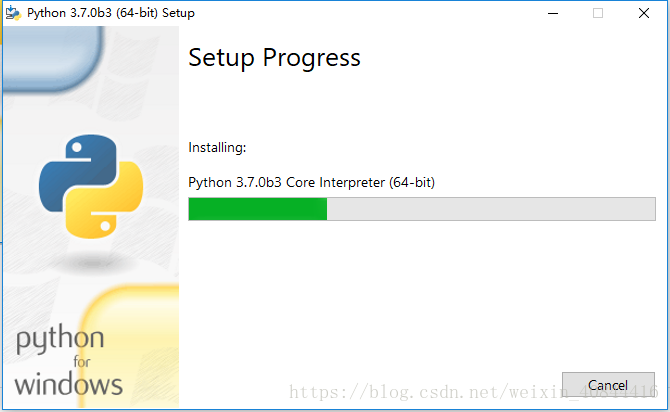


#### 1.4选择一个自己喜欢的安装位置

点击Install开始安装

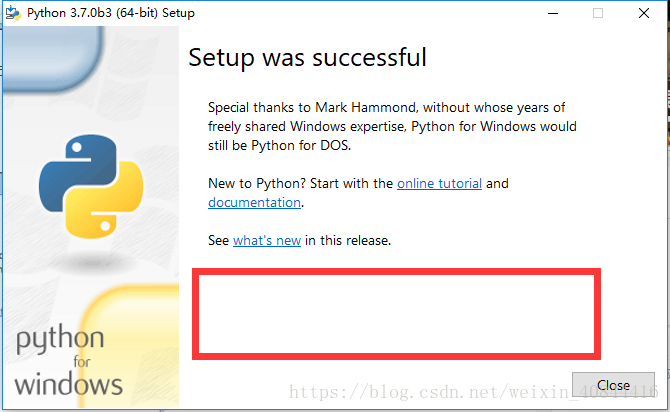


#### 1.5等待进度条加载完毕



#### 1.6安装完毕，点击Close关闭

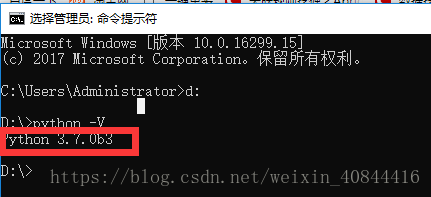
若方框位置出现管理员字样则点击授权再关闭



#### 1.7验证：运行cmd

进入到自己的安装目录下，运行语句：python -V，若环境变量已配置成功，则任何目录都能成功运行python -V，请看步骤1.2

若显示出Python对应版本则表示安装成功

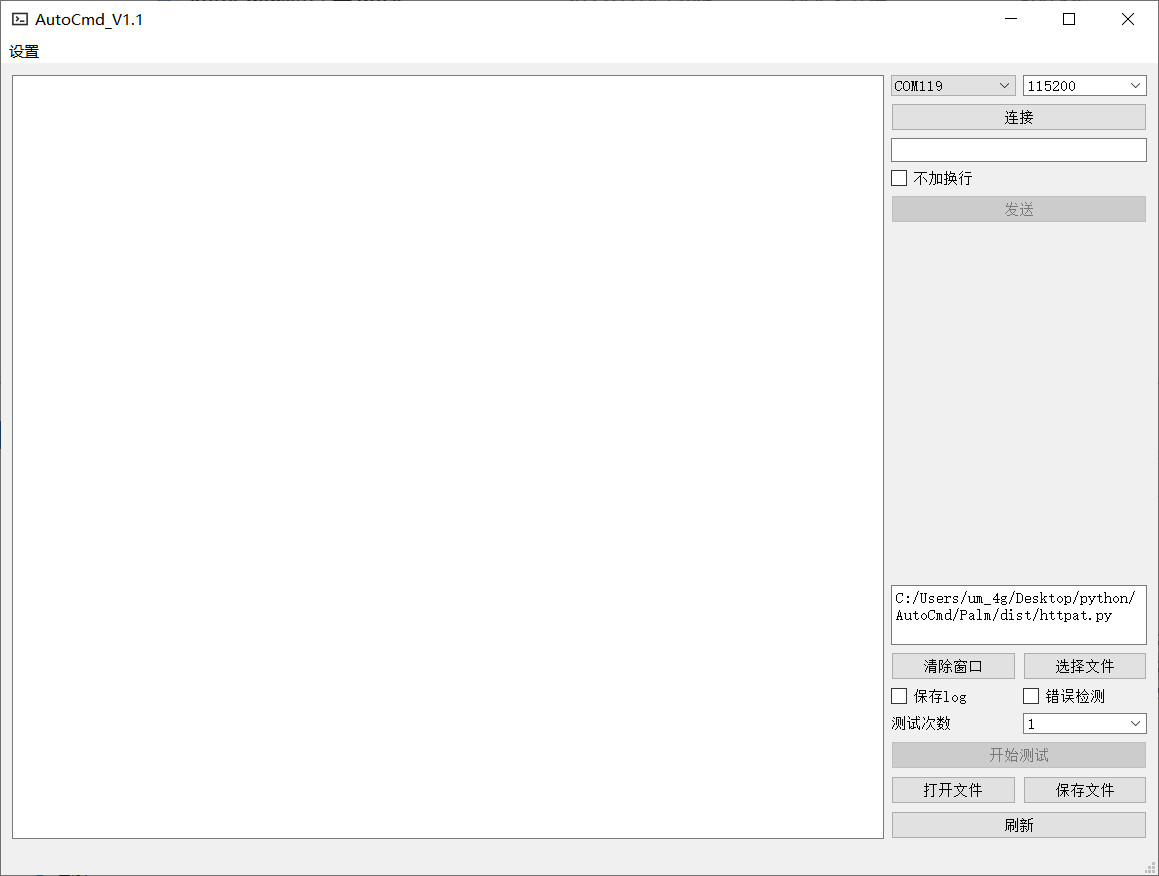


## 工具说明



运行AutoCMD\_nlog.exe,其中AutoCMD\_log.exe可用于脚本编写时测试使用

打开界面如下：



按键说明：

连接：进行连接com口

发送：发送AT命令，默认带\r\n

清除窗口：将显示界面清除

选择文件：选择测试脚本

刷新：刷新com列表

## 脚本编写

参考httpat.py脚本



列表元组规定：

{

'cmd': 'ATE0', #命令名称，对于全局变量可用{{}}引用

'fun': fun, #命令处理函数

'succ': 'OK', #成功结束命令标志，不定义，默认‘OK’

'err': 'ERROR',#错误结束命令标志,不定义，默认‘ERROR’

'quit': 'ERROR',#错误结束命令并退出测试标志，不定义，不进行检测处理

'time': 2,#延迟2秒发送该命令

'jump':-1#命令跳转，0为当前命令，-1为上一条命令，1为下一条指令，要结合自定义fun返回值jump使用，不定义默认为0

}

处理函数规定：

def fun(device,data,index)

device.print() :在log打印，在log文件中可以看出

device.Send(str):发送字符串str,

命令模式： +cmd:发送内容

+cmdz:发送内容,会自动发送ctrl+z

+exit:time,index :经过time秒后，测试第index条命令

+urc:数据打印在AutoCMD工具中

data :为device返回的一行结果

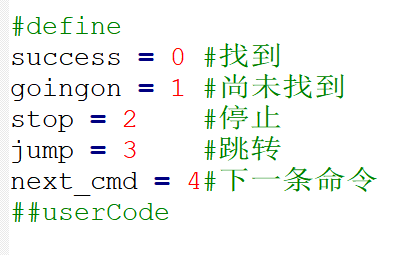
index :表示当前执行命令的index

## 脚本api模块

api模块：封装处理函数，代码位于./config/Palm-cmd/api.py

在自定义脚本使用时，需要用api.xxx,可调用对应内容

定义以下返回值：



### 1、fuc\_handle--命令解析

#check 元组，解析获得的字符串

#ret\_list返回查找到符合条件的字符串列表，ret\_list必须为列表：ret = []

#send 找到符合字符串之后发送数据，以'+cmd:','+cmdz:','+urc:'开头

#例如target = '+COPS:',check:{0:'0',1:'2',2:"46011",3:'7'}

#找到返回success，找不到返回goingon

def fuc\_handle(device,data,target = None, check = None, ret\_list=None, send=None)

demo:

1. ##AT+COPS?
2. **def** fun4(device,data,index) :
3. #+COPS: 0,2,"46011",7
4. \_ret\_list = []
5. ret = api.fuc\_handle(device,data,target='+COPS:',check={3:'7'},ret\_list = \_ret\_list)
6. **if** ret==api.success:
7. **return** api.success
8. **else**:
9. **return** api.fuc\_retry(device,data,5)

### 2、fuc\_retry--命令循环

#命令重新测试，retry\_times:测试次数

#未满足循环条件返回goingon，进行循环返回jump，循环次数到了返回stop

def fuc\_retry(device,data,retry\_times,succ='OK',err='ERROR')

### 3、fuc\_get\_line--命令获取特定行

#获取从发送命令开始的特定行数据 ，lines必须为列表：lines= []

#获取成功，返回success，未获取成功返回goingon

def fuc\_get\_line(device,data,lines)

### 4、fuc\_cmd--发送AT命令

#获取从发送AT命令

#无返回值

def fuc\_cmd(device,cmd)

### 5、fuc\_cmdz--发送数据后带ctrl+z

#获取从发送命令带ctrl+z

#无返回值

def fuc\_cmdz(device,data)

### 6、fuc\_reset--重新测试

#重新断开，经过time秒后重新连接，从index开始测试

#无返回值

def fuc\_reset(device,time,index)