PowerShell 学习笔记

Cheng Jun

2018 年 9 月 29 日

目录

1 前言 **2**
- 1.1 笔记作者 . . . 2
- 1.2 学习目的 . . . 2
- 1.3 准备工作 . . . 2
  - 1.3.1 安装 Python+Jupyter Lab . . . 2
  - 1.3.2 安装 PowerShell Kernel . . . 2
- 1.4 学习环境 . . . 2
  - 1.4.1 系统自带的 Powershell . . . 2
  - 1.4.2 Powershell Core . . . 3
- 1.5 编辑和运行命令 . . . 4
  - 1.5.1 启动 Powershell . . . 4
  - 1.5.2 以管理者权限模式启动 Powershell . . . 5
  - 1.5.3 输入命令 . . . 6
  - 1.5.4 在 VS Code 中编辑和运行命令 . . . 6
- 1.6 参考材料 . . . 7

2 基础 **7**
- 2.1 帮助 . . . 7
- 2.2 别名 . . . 8
- 2.3 清理屏幕 . . . 9

3 文件操作 **9**
- 3.1 设置路径 . . . 9
- 3.2 生成文件和文件夹 . . . 9
- 3.3 删除文件 . . . 9
- 3.4 管道操作 . . . 10

4 对象 **11**
4.1 选择对象属性 11
4.2 对象排序 12
4.3 筛选 13
4.3.1 直接筛选 13
4.3.2 使用-filter 13
4.3.3 Where-Object 13

5 变量 **14**

6 判断 **15**

7 循环 **16**

8 执行外部命令 **18**
8.1 当前目录下 18
8.2 运行全局的命令 18

1 前言

1.1 笔记作者

实证研究小青年。

日常研究和关注经济、金融与会计等领域的问题，主要采用计量经济学和其他数据分析手法撰写学术论文和研究报告。

研究之余，泛读文史哲，关注自由与开源动态。在日常研究和工作中，喜欢使用最新的软件工具，并且喜欢选用自由和开源工具。

邮箱：cheng081@qq.com

Github：https://github.com/chengjun90

欢迎交流。

1.2 学习目的

不是完整的掌握 PowerShell，而是了解基本的 PowerShell 的应用，方便在研究工作和生活中需要的时候用一点脚本。

比如：

- Data Science at the Command Line
- Command Line Tricks For Data Scientists

1.3 准备工作

1.3.1 安装 Python+Jupyter Lab

这个就不说了，很好安装。直接搜索。

1.3.2 安装 PowerShell Kernel

可以使用这里 https://github.com/vors/jupyter-powershell 提供的包。

```
pip install powershell_kernel
python -m powershell_kernel.install
```

这样就可以在 Jupyter Lab 中学习和使用 Powershell 了。

1.4 学习环境

1.4.1 系统自带的 Powershell

这里是 Powershell 版本是 5.1。

```
In [1]: $psversiontable
```

```
Name                           Value
----                           -----
PSVersion                      5.1.17134.228
PSEdition                      Desktop
PSCompatibleVersions           {1.0, 2.0, 3.0, 4.0…}
BuildVersion                   10.0.17134.228
CLRVersion                     4.0.30319.42000
WSManStackVersion              3.0
PSRemotingProtocolVersion      2.3
SerializationVersion           1.1.0.1
```

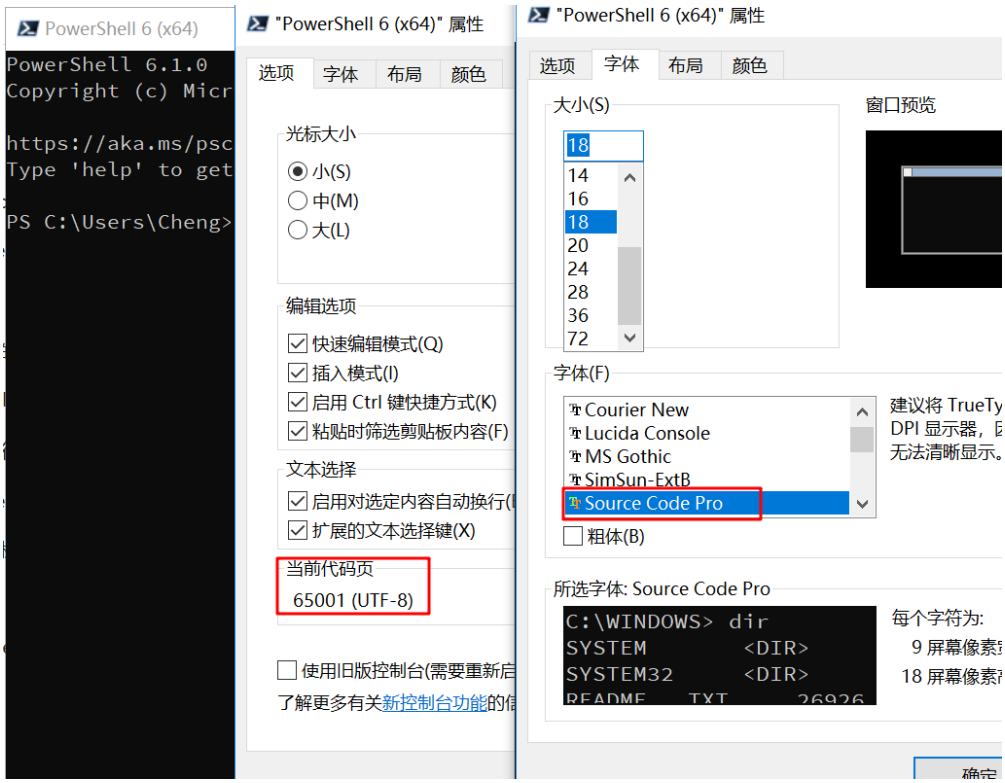
1.4.2 Powershell Core

当然也可以自行安装跨平台的Powershell Core。

```
PS C:\Users\Cheng> $psversiontable
```

```
Name                           Value
----                           -----
PSVersion                      6.1.0
PSEdition                      Core
GitCommitId                    6.1.0
OS                             Microsoft Windows 10.0.17134
Platform                       Win32NT
PSCompatibleVersions           {1.0, 2.0, 3.0, 4.0...}
PSRemotingProtocolVersion      2.3
SerializationVersion           1.1.0.1
WSManStackVersion              3.0
```

2018 年，Powershell Core 已经支持 utf-8 编码了，可以使用第三方字体，例如Source Code Pro。



1.5 编辑和运行命令

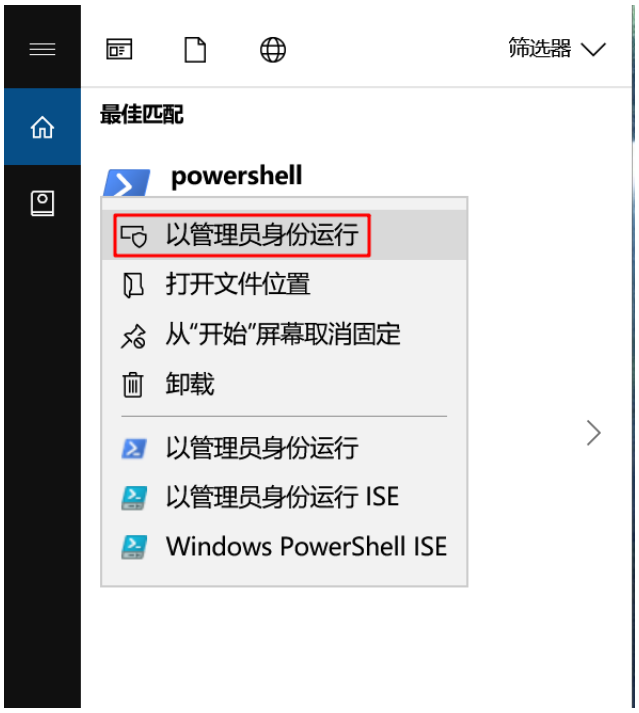
1.5.1 启动 Powershell

直接在窗口输入“Powershell”，然后点击。

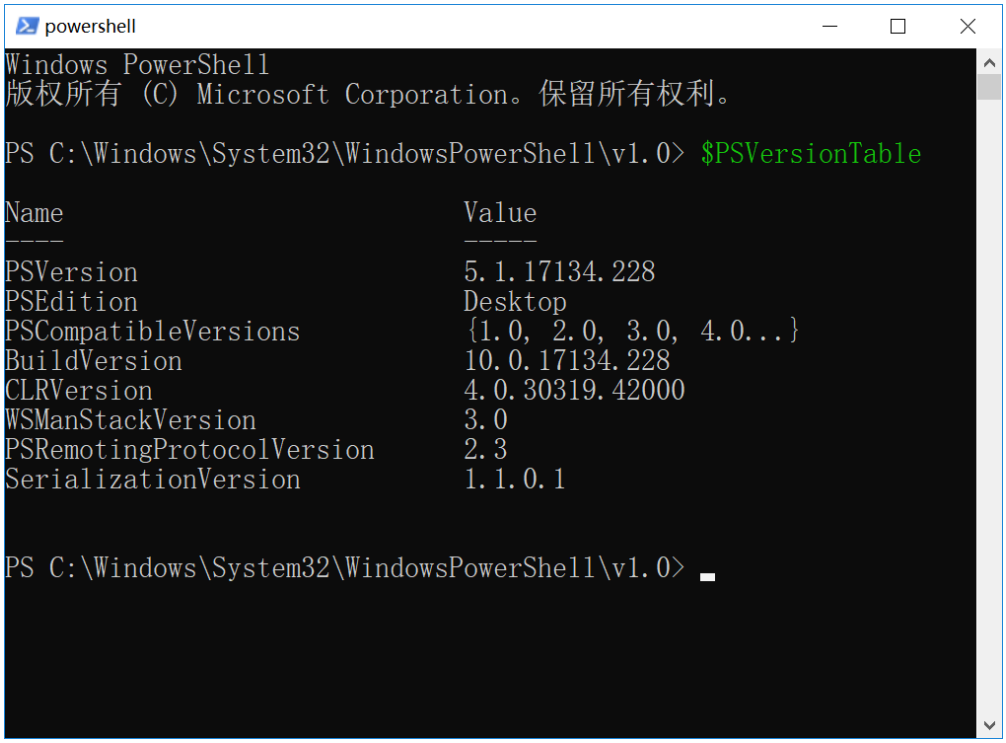
如果是想调用 Powershell Core，那么就输入“pwsh”。

1.5.2 以管理者权限模式启动 Powershell

例如在按照 Python 包的时候，常常需要进入这种模式。



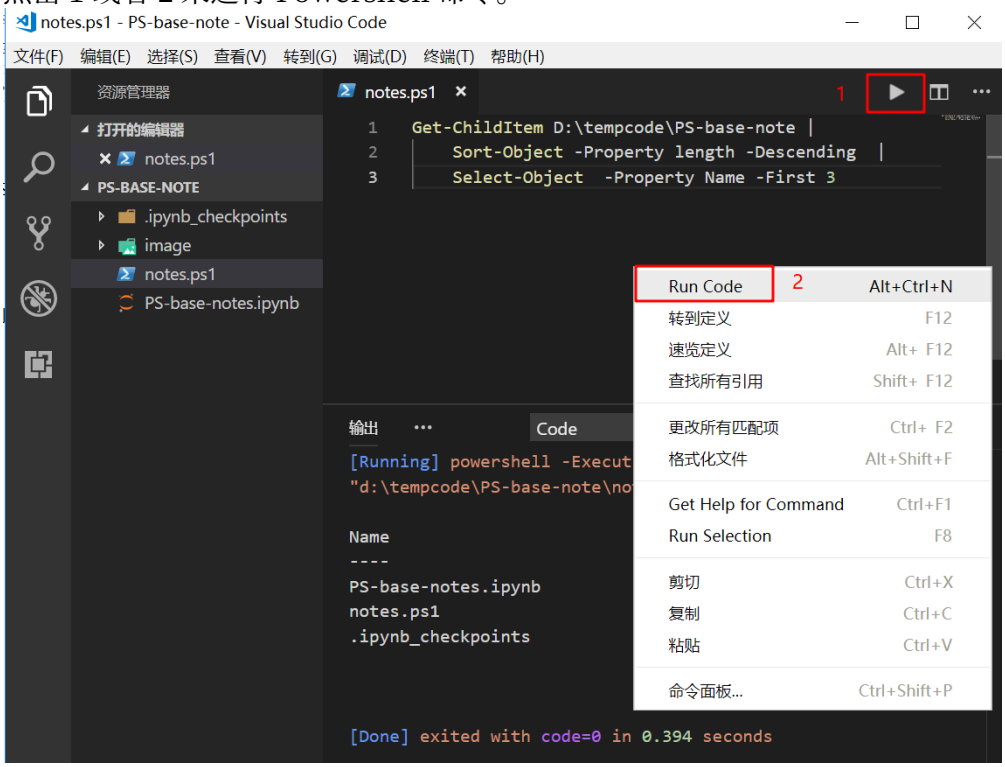
1.5.3 输入命令



1.5.4 在 VS Code 中编辑和运行命令

Powershell 文件的类型名称是 ps1。

点击 1 或者 2 来运行 Powershell 命令。



1.6 参考材料

- 微软 PowerShell 材料
- Learn Windows PowerShell in a Month of Lunches

2 基础

2.1 帮助

善于使用 Get-Help 命令来查找帮助文件

```
In [2]: Get-Help Get-ChildItem


■■■    Get-ChildItem

■■■
    Get-ChildItem [[-Path] <string[]>] [[-Filter] <string>]  [<CommonParameters>]

    Get-ChildItem [[-Filter] <string>]  [<CommonParameters>]
```

```
■■■
    gci
    ls
    dir


■■■
    Get-Help ■■■■■■■■■■■cmdlet ■İ■■■■■■■■■■■'■■■■■■■■
        -- ■Ç■■■■■■■■■■■■■cmdlet ■■■■■■■■■■■■■■'■ Update-Help■■
        -- ■Ç■■■■■ □ ■■cmdlet ■İ■■■■■■■■■■: "Get-Help Get-ChildItem -Online" ■■
           ■■■ https://go.microsoft.com/fwlink/?LinkID=113308■■
```

有的时候命令记不全，那么可以使用 Tab 键来补全命令。

或者使用 Show-Command 命令来帮助自己。Show-Command 可以弹出窗口来选择命令、填写命令参数。

2.2 别名

Powershell 的命令很多有别名，可以理解成命令的简短昵称，便于输入命令。

```
In [3]: Get-Alias -Definition "Set-Location"

CommandType     Name                                               Version    Sourc
-----------     ----                                               -------    -----
Alias           cd -> Set-Location
Alias           chdir -> Set-Location
Alias           sl -> Set-Location


In [4]: Get-Alias -Definition "Get-ChildItem"
```

```
CommandType     Name                                               Version    Sourc
-----------     ----                                               -------    -----
Alias           dir -> Get-ChildItem
Alias           gci -> Get-ChildItem
Alias           ls -> Get-ChildItem
```

2.3 清理屏幕

直接使用 cls 命令即可。

3 文件操作

3.1 设置路径

比如我们常常需要切换工作路径。这个时候需要使用 Set-Location 命令。
不过我常常使用其别名 cd。

```
cd D:\tempcode
Set-Location D:\tempcode
```

3.2 生成文件和文件夹

mkdir 可以直接生成文件夹。

```
mkdir test
```

通用的可以使用 New-Item。

```
New-Item "D:\tempcode\ps code" -Type Directory
New-Item "D:\tempcode\ps code\note.txt" -Type File
```

3.3 删除文件

通用的可以使用 New-Item。
删除文件:

```
Remove-Item "D:\tempcode\ps code\note.txt"
```

删除整个文件夹和子文件:

```
Remove-Item "D:\tempcode\ps code" -recurse
```

3.4 管道操作

PowerShell 通过管道（pipeline）把命令互相连接起来。

$f_1(x_1) \to f_2(x_2) \to f_3(x_3)$

这样就可以把命令连接起来。

对指定文件夹下面的全部文件，按照文件大小降序排列，并且列示前 10 个文件的文件名。

```
In [5]: dir D:\tempcode -Recurse |
        Sort-Object -Property length -Descending  |
        Select-Object  -Property Name -First 10
```

```
Name
----
test.exe
PS-base-notes.pdf
ps4-run.png
ps5-core.png
ps3-code.png
ps1.png
ps2-ad.png
PS-base-notes.tex
PS-base-notes.log
PS-base-notes-checkpoint.ipynb
```

命令结果还可以导出到文件中去。

```
Dir > DirectoryList.txt
```

或者是

```
Dir |
Out-File DirectoryList.txt
```

4 对象

有了对象之后，就有对象的标签（属性）和对象的行为（方法）。

4.1 选择对象属性

这个主要是 Select-Object 命令。

```
In [6]: cd D:\tempcode
        ls | Select-Object -property Name,Length
```

```
cd D:\tempcode
ls | Select-Object -property Name,Length

Name                Length
----                ------
.ipynb_checkpoints
.vscode
ps code
PS-base-note
test
test-site
1-query-pq.py       517
2-create-table.py   1139
data.json           516
DirectoryList.txt   5634
jsondata.py         463
juliadoc.py         470
learn.py            53
tempcode.Rproj      248
test.cpp            104
test.exe            3071535
test.txt            145
testFolder          0
txt-line-number.py  503
Untitled.ipynb      1013
■■□■.R             244
■■■■■■sql          46
```

4.2 对象排序

这里主要用到 Sort-Object。

```
In [7]: ls | Sort-Object -property Name,Length
```

```
    L¼: D:\tempcode


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        2018/9/29     14:05                .ipynb_checkpoints
d-----        2018/9/25     22:14                .vscode
-a----        2018/9/19     16:03            517 1-query-pq.py
-a----        2018/9/18     21:16           1139 2-create-table.py
-a----        2018/9/21      9:28            516 data.json
-a----        2018/9/29     10:26           5634 DirectoryList.txt
-a----        2018/9/21      9:27            463 jsondata.py
-a----        2018/9/19     20:24            470 juliadoc.py
-a----        2018/9/26      9:48             53 learn.py
d-----        2018/9/29     15:42                ps code
d-----        2018/9/29     16:13                PS-base-note
-a----        2018/9/15     20:06            248 tempcode.Rproj
d-----        2018/9/29      9:58                test
-a----        2018/9/29     15:49            104 test.cpp
-a----        2018/9/29     15:49        3071535 test.exe
-a----        2018/9/23     11:43            145 test.txt
-a----        2018/9/29      9:56              0 testFolder
d-----        2018/9/18     15:52                test-site
-a----        2018/9/23     12:00            503 txt-line-number.py
-a----        2018/9/27     16:33           1013 Untitled.ipynb
```

```
-a----         2018/9/19     16:03             46 ■■■■■■sql
-a----         2018/9/15     20:12            244 ■■□■.R
```

4.3 筛选

4.3.1 直接筛选

按照文件名称进行筛选文件。

```
ls -name *.py
ls -name t*.py
ls -name *.py,*.cpp
```

或者是-Include 参数，可以实现对文件按照类型名称进行筛选。

```
Get-ChildItem D:\tempcode -Recurse -Include *.py,*.ipynb
```

4.3.2 使用-filter

```
In [8]: ls -filter "Name -like '*.py'"
```

4.3.3 Where-Object

更复杂的情况，可以使用 Where-Object（别名为 Where）来筛选。
这个时候需要用到比较运算符:

- -eq，相等
- -ne，不等于
- -ge 和-le，大于或等于，小于或等于
- -gt 和-lt，大于和小于

如果需要区分字符大小写，可以使用:-ceq，-cne，-cgt，-clt，-cge，-cle。
布尔运算符: - -and - -or - -not

```
PS D:\tempcode>  (12 -gt 8) -and (12 -ne 12)
False
```

比较文本字符串时，还有几个常用的比较运算符：

- -like，接受 * 作为一个通配符，来比较字符，忽略大小写
- -notlike，忽略大小写
- -clike，区分大小写
- -cnotlike，区分大小写
- -match，正则表达式匹配，忽略大小写
- -notmatch，忽略大小写
- -cmatch，区分大小写
- -cnotmatch，区分大小写

```
In [9]: dir D:\tempcode |
        Where-Object -filter {$_.Name -like "*test*"} |
        Select-Object -Property Name


Name
----
test
test-site
test.cpp
test.exe
test.txt
testFolder
```

5 变量

变量可以直接生成。

Remove-Variable 可以删除变量。

```
PS D:\tempcode> $ind="C28"
PS D:\tempcode> $ind
C28
PS D:\tempcode> $roa=0.12
PS D:\tempcode> $roa
```

```
0.12
PS D:\tempcode> Remove-Variable ind,roa
PS D:\tempcode> $ind
PS D:\tempcode>
```

6 判断

- IF-ELSEIF-ELSE
- Switch

```
In [10]: $value=0
         If( $value -eq 1 )
         {
             "A"
         }
         Elseif( $value -eq 2)
         {
             "B"
         }
         Else
         {
             "C"
         }
```

```
$value=0
If( $value -eq 1 )
>> {
>>     "A"
>> }
>> Elseif( $value -eq 2)
>> {
>>     "B"
>> }
>> Else
>> {
>>     "C"
>> }
>>
```

```
C
```

```
In [11]: $value=1
         switch($value)
         {
             1 {"A"}
             2 {"B"}
             3 {"C"}
         }
```

```
$value=1
switch($value)
>> {
>>     1 {"A"}
>>     2 {"B"}
>>     3 {"C"}
>> }
>>
A
```

7 循环

- ForEach-Object
- Foreach
- Do While
- For
- Switch

```
In [12]: dir D:\tempcode\PS-base-note | ForEach-Object {$_.Name}
```

```
.ipynb_checkpoints
image
notes.ps1
PS-base-notes.ipynb
PS-base-notes.log
PS-base-notes.pdf
```

PS-base-notes.tex

```
In [13]: $x=1..3
         foreach ($i in $x)
         {
             $i*2
         }
```

```
$x=1..3
foreach ($i in $x)
>> {
>>     $i*2
>> }
>>
2
4
6
```

```
In [14]: for($year=2014; $year -le 2018; $year ++)
         {
             "year"+$year
             -Join("year-",$year)
         }
```

```
year2014
year-2014
year2015
year-2015
year2016
year-2016
year2017
year-2017
year2018
year-2018
```

8 执行外部命令

8.1 当前目录下

例如编写一个简单的 C++ 代码，然后编译成 exe 文件。

```
#include <iostream>

using namespace std;

int main()
{
cout << "code 123" << endl;
return 0;
}
```

```
PS D:\tempcode>  .\test.exe
code 123
PS D:\tempcode> .\test
code 123
PS D:\tempcode>
```

8.2 运行全局的命令

就是类型为 exe 的文件路径在 Path 环境变量中，例如 Julia.exe 和 R.exe。
下面演示如何进入 Julia 和 R，如何退出 Julia 和 R。

```
PS D:\tempcode> julia
               _
   _       _ _(_)_     |  Documentation: https://docs.julialang.org
  (_)     | (_) (_)    |
   _ _   _| |_  __ _   |  Type "?" for help, "]?" for Pkg help.
  | | | | | | |/ _` |  |
  | | |_| | | | (_| |  |  Version 1.0.0 (2018-08-08)
 _/ |\__'_|_|_|\__'_|  |  Official https://julialang.org/ release
|__/                   |

julia> exit()
PS D:\tempcode> julia.exe
```

```
               _
   _       _ _(_)_     |  Documentation: https://docs.julialang.org
  (_)     | (_) (_)    |
   _ _   _| |_  __ _   |  Type "?" for help, "]?" for Pkg help.
  | | | | | | |/ _` |  |
  | | |_| | | | (_| |  |  Version 1.0.0 (2018-08-08)
 _/ |\__'_|_|_|\__'_|  |  Official https://julialang.org/ release
|__/                   |

julia> exit()
PS D:\tempcode> R.exe

R version 3.5.1 (2018-07-02) -- "Feather Spray"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R

'license()''licence()'

R.
'contributors()'
'citation()'RR

'demo()''help()'
'help.start()'HTML
'q()'R.

Microsoft R Open 3.5.1
The enhanced R distribution from Microsoft
Microsoft packages Copyright (C) 2018 Microsoft Corporation

Using the Intel MKL for parallel mathematical computing (using 4 cores).

Default CRAN mirror snapshot taken on 2018-08-01.
See: https://mran.microsoft.com/.
```

```
> q()
是否保存工作空间映像? [y/n/c]: n
PS D:\tempcode>
```