

目录

前言	1.1
概述	1.2
通用知识	1.3
文件和文件夹	1.3.1
显示文件后缀名	1.3.1.1
终端shell	1.3.2
软件安装和包管理器	1.3.3
心得	1.3.3.1
IP地址	1.3.4
Windows	1.4
Mac	1.5
brew	1.5.1
app安全限制	1.5.2
Linux	1.6
附录	1.7
参考资料	1.7.1

计算机电脑知识总结

- 最新版本： v0.5
- 更新时间： 20200924

简介

总结电脑计算机方面的常用基础知识，包括多个系统中通用知识，比如显示文件名后缀，终端shell，软件安装和包管理器，以及对应心得，IP地址，以及其他系统，如Windows，和Mac，及其中的brew如何加代理加速下载，如何解除app使用的安全限制等，以及Linux系统常用心得。

源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

Gitbook源码

- [crifan/computer_tech_summary](#): 计算机电脑知识总结

如何使用此Gitbook源码去生成发布为电子书

详见：[crifan/gitbook_template: demo how to use crifan gitbook template and demo](#)

在线浏览

- [计算机电脑知识总结 book.crifan.com](#)
- [计算机电脑知识总结 crifan.github.io](#)

离线下载阅读

- [计算机电脑知识总结 PDF](#)
- [计算机电脑知识总结 ePUB](#)
- [计算机电脑知识总结 Mobi](#)

版权说明

此电子书教程的全部内容，如无特别说明，均为本人原创和整理。其中部分内容参考自网络，均已备注了出处。如有发现侵犯您版权，请通过邮箱联系我 `admin 艾特 crifan.com`，我会尽快删除。谢谢合作。

鸣谢

感谢我的老婆陈雪的包容理解和悉心照料，才使得我 `crifan` 有更多精力去专注技术专研和整理归纳出这些电子书和技术教程，特此鸣谢。

更多其他电子书

本人 crifan 还写了其他 100+ 本电子书教程，感兴趣可移步至：

[crifan/crifan_ebook_readme: Crifan的电子书的使用说明](#)

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2021-01-16
22:14:21

概述

之前作为普通用户使用电脑期间，和作为技术开发人员使用电脑做技术开发期间，用过各种电脑系统。

此处整理出各种相对通用的基础知识和使用经验，供参考。

子教程

其中部分内容可参考相关独立教程：

- 投屏
 - 【整理Book】手机电脑投屏方案总结
- 音视频
 - 【整理Book】电脑中音视频文件播放与处理总结
- 文件内容对比
 - 【整理Book】文件内容对比工具总结
- 画图
 - 好用的画图工具
 - https://book.crifan.com/books/best_diagram_tool/website/
- 编辑器和IDE
 - 编辑器和IDE总结
 - https://book.crifan.com/books/editor_ide_summary/website/

以及：

- PC端
 - Linux
 - 【整理Book】Linux使用和开发心得
 - Mac
 - 苹果相关开发总结
 - https://book.crifan.com/books/apple_develop_summary/website/
- 移动端
 - 前言 · 移动端APP开发总结
 - https://book.crifan.com/books/mobile_app_summary/website/

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-08-09
10:14:59

通用知识

此处整理各种系统中都会涉及到一些通用知识。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09
10:14:59

文件和文件夹

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09
10:14:59

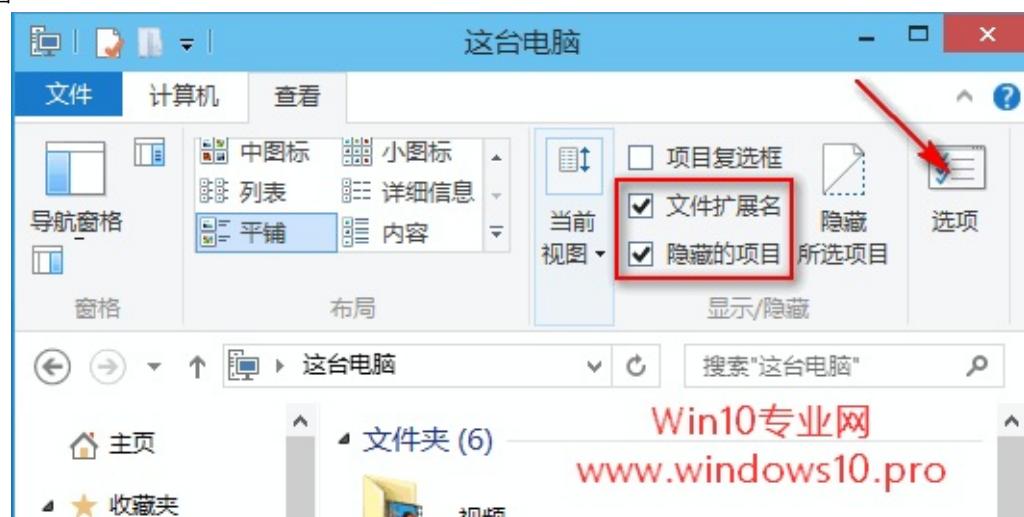
显示文件后缀名

关于如何设置去 显示文件后缀名 :

- Windows

- Win10

- 文字: 资源管理器 -> 查看 ->勾选 或 取消勾选 文件扩展名
 - 截图



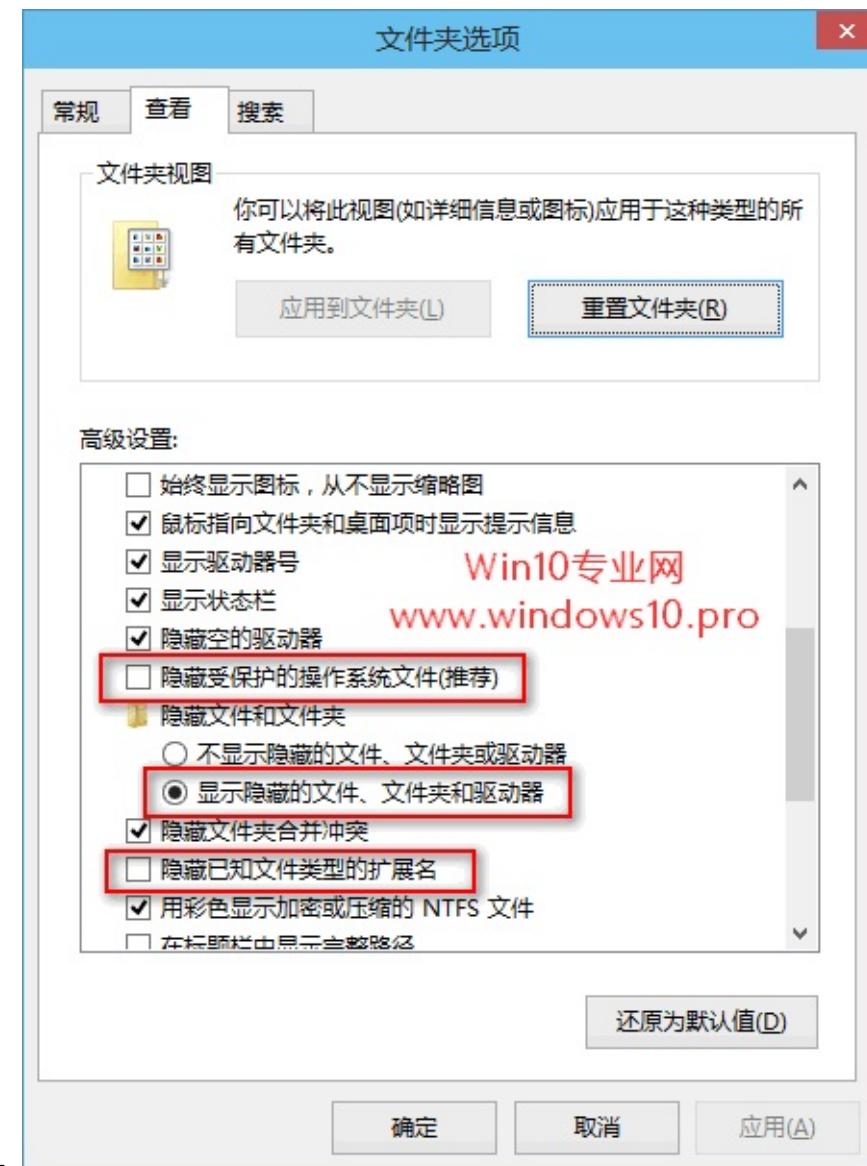
- 进一步

- 想要显示被隐藏的操作系统文件

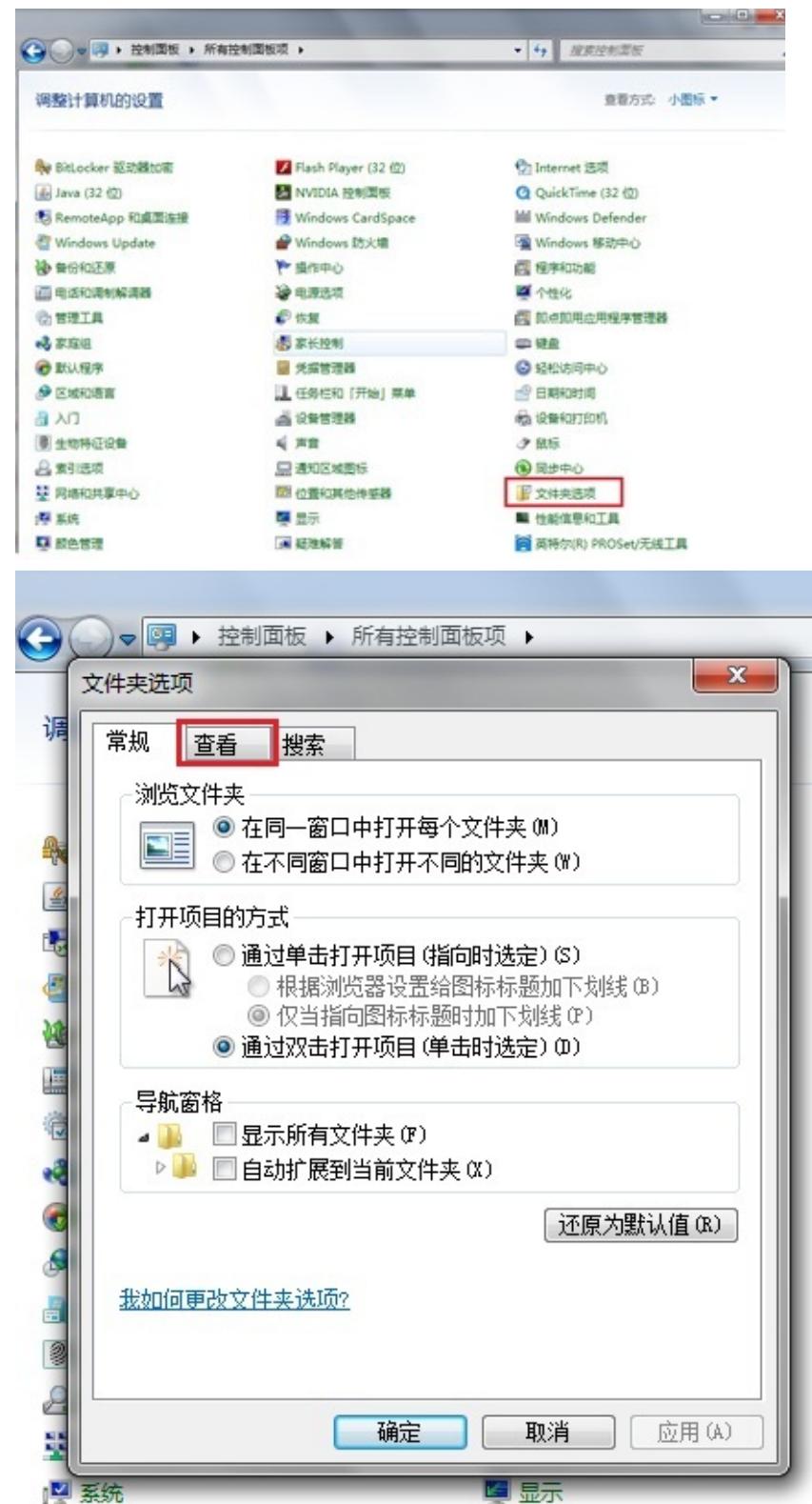
- 文字

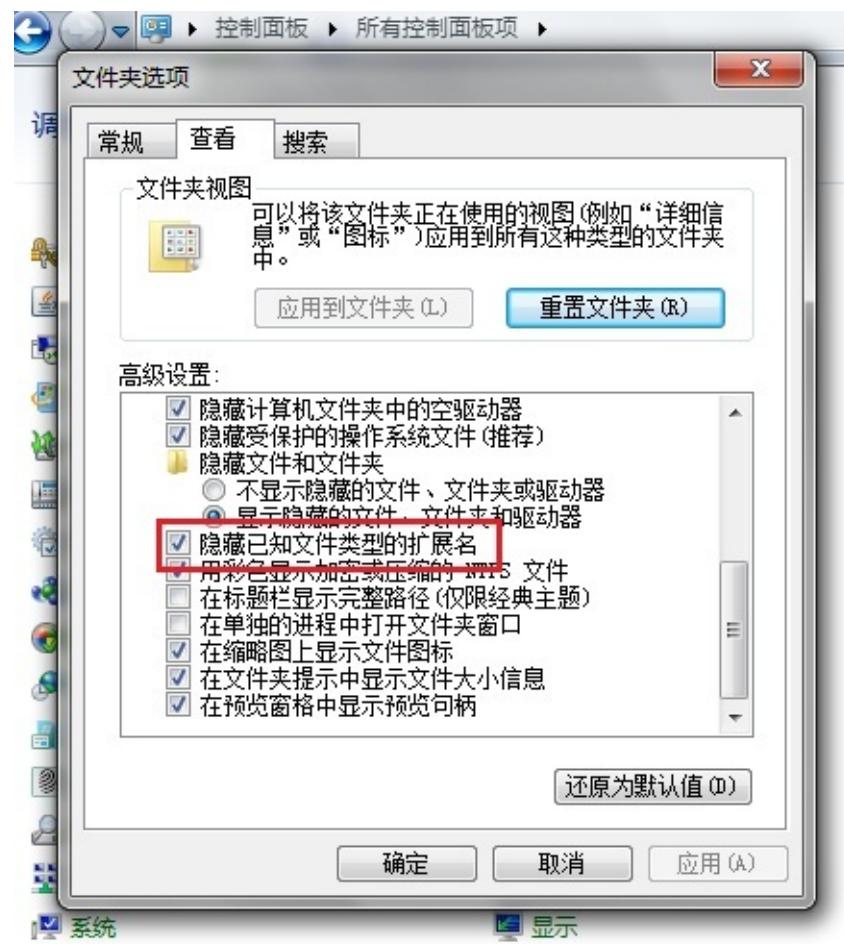
- 勾选 文件扩展名 和 隐藏的项目 即可显示文件扩展名和普通的隐藏文件、文件夹或驱动器
 - 但是还不能显示被隐藏的操作系统文件
 - 点击 选项 图标打开 文件夹选项 窗口, 切换到 查看 选项卡
 - 在下面的 高级设置 中取消勾选 隐藏受保护的操作系统文件
 - 同时你会发现刚才所做的勾选 文件扩展名 和 隐藏的项目 的操作在这里同样可以通过选中 显示隐藏的文件、文件夹或驱动器 和 取消勾选 隐藏已经文件类型的扩展名 来实现。

- 截图:



- Win7
 - 文字
 - 控制面板 -> 文件夹选项 -> 查看 -> 高级设置 -> 隐藏已知文件类型的扩展名
 - 截图

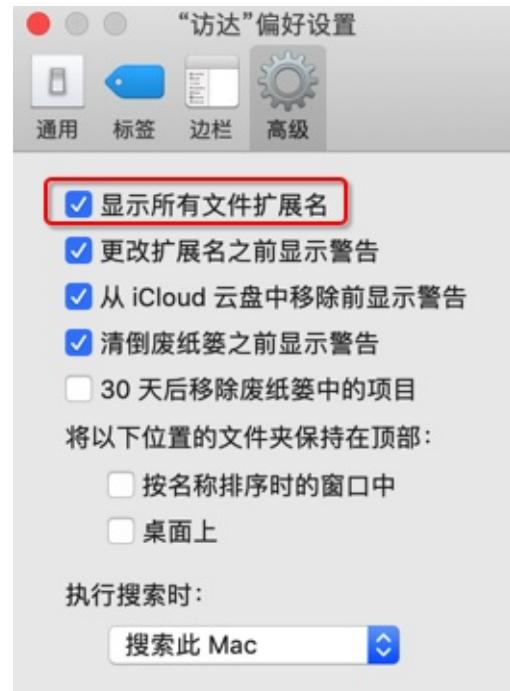




- Mac

- 如何设置是否显示文件名后缀
 - 文字：访达 Finder -> 偏好设置 -> 高级 -> 勾选或取消勾选 显示所有文件扩展名
 - 截图：





- 举例

- 某png图片是否显示文件后缀名
- 显示



- 不显示



- 另外

- 请注意此处 种类是： PNG图像
- 表示是后缀是 .png 的图像文件=照片

- 更多解释

- 在 Mac 上显示或隐藏文件扩展名 - Apple 支持
- <https://support.apple.com/zh-cn/guide/mac-help/mchlp2304/10.14/mac/10.14>

终端shell

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09
10:14:59

软件安装和包管理器

普通用户安装软件

作为电脑的普通用户，典型的安装软件方式是：

- Windows
 - 下载安装包exe文件，双击，根据提示，一步步去安装
 - 下载了免安装的，解压后直接使用，无需安装。
- Linux
 - 方式1：下载 源码 自己 编译 和 安装
 - 方式2：用各种 包管理器 去 install 安装

不同系统，有各自常用的包管理器，去用于安装软件：

包管理器

不同系统的常见包管理器

- 系统
 - Mac
 - Homebrew
 - brew install xxx
 - MacPorts
 - port install xxx
 - Linux
 - Debian / Ubuntu / Termux
 - apt
 - apt-get install xxx
 - Red Hat / CentOS
 - yum
 - yum install xxxx
 - FreeBSD / Termux
 - pkg install xxx

其中Linux由于有很多发行版，比较特殊，所以专门再详细解释：

Linux系统中的包管理器

Linux有很多不同发行版，每个发行版基本都有自己的用于安装软件的 包管理器 = 包管理系统 = package management tool。常见的一些有：

- Debian / Ubuntu：
 - 软件包格式： deb
 - 包管理工具： dpkg
 - 前端工具： apt
- Fedora / Red Hat Enterprise Linux / CentOS / openSUSE / Mandriva Linux / Mageia
 - 软件包格式： rpm
 - 前端工具

- Fedora : dnf
- Red Hat Enterprise Linux / CentOS : yum
- openSUSE : Zypp
- Mandriva Linux / Mageia : urpmi

举例

- vlc
 - Debian / Ubuntu / Termux
 - apt-get install vlc
 - Red Hat / CentOS
 - yum install vlc
 - FreeBSD / Termux
 - pkg install vlc
- sshfs
 - CentOS
 - yum install sshfs
 - Fedora 22+
 - dnf install sshfs
 - Debian / Ubuntu
 - sudo apt-get install sshfs

补充说明:

- 如果安装期间缺少权限，则往往要加上su或sudo以管理员身份才能安装
 - 举例

```
sudo apt-get install vlc
sudo yum install vlc
```

各种编程语言的常见包管理器

- 编程语言
 - Python
 - pip
 - 典型用法:
 - pip install xxx
 - 私有源
 - pypiserver
 - pypi
 - pipenv
 - pipenv install xxx
 - 配置文件: Pipfile
 - Mac 的 iOS : Swift / Objective-C
 - Carthage
 - carthage update
 - 配置文件: Cartfile
 - Cocoapods
 - pod install
 - 配置文件: Podfile
 - Javascript
 - npm

- 用法: `npm install xxx`
- 配置文件: `package.json`
- `yarn`
 - `yarn add xxx`
- `Bower`
 - `bower install xxx`
- `PHP`
 - `composer`
 - `composer install`
 - 配置文件: `composer.json`
- `Java`
 - `Maven`
 - 典型用法
 - 配置文件: `pom.xml`
 - 安装:
 - `mvn install`
 - 私有仓库=私服=公司内部的源
 - `Nexus`
 - `gradle`
 - 配置文件: `build.gradle`
 - 运行: `gradle xxx`
 - `Ruby`
 - `gem`
 - 配置文件: `Gemfile`
 - `.Net`
 - `NuGet`
 - 配置文件: `nupkg`

补充说明:

关于镜像和源:

- 综合=全能
 - `GitHub Package Registry`
 - 支持:
 - `JavaScript` : `npm`
 - `Java` : `Maven`
 - `Ruby` : `RubyGems`
 - `.NET` : `NuGet`
 - Docker镜像
 - 将要支持:
 - `Python`
 - `PHP`

包管理器的心得

相同包不同系统名字略有不同

同一个工具和包，不同的系统中，名字往往不太一样，但往往是类似的规律

比如

某个库的开发的包的命令规律：

- Ubuntu
 - libxxxx-dev
 - apache2-dev
- CentOS
 - xxx-devel

举例

openssl

openssl 这个库，对应的（用于开发，被别人源码编译期间所需要的）开发包：

- Ubuntu (Debian , 基于 Debian 的树莓派) : libssl-dev
 - 安装: apt-get install libssl-dev
- CentOS : openssl-devel
 - 安装: yum install openssl-devel

zeromq

[ZeroMQ | Download](#)

- CentOS7
 - yum install -y zeromq-devel
- Debian / Ubuntu
 - apt-get install libzmq3-dev
- Fedora
 - dnf install zeromq-devel
- Arch
 - pacman -S zeromq
- SUSE
 - zypper install zeromq-devel

policycoreutils-python

- Debian : apt-get install policycoreutils-python-utils
- Ubuntu : apt-get install policycoreutils-python-utils
- Kali Linux : apt-get install policycoreutils-python-utils
- CentOS : yum install policycoreutils-python
- Fedora : dnf install policycoreutils-python-utils
- Raspbian : apt-get install policycoreutils-python-utils

○ Debian	<code>apt-get install policycoreutils-python-utils</code>
● Ubuntu	<code>apt-get install policycoreutils-python-utils</code>
⚡ Kali Linux	<code>apt-get install policycoreutils-python-utils</code>
✿ CentOS	<code>yum install policycoreutils-python</code>
ƒ Fedora	<code>dnf install policycoreutils-python-utils</code>
aspberry Raspbian	<code>apt-get install policycoreutils-python-utils</code>

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2021-01-16
21:30:25

IP地址

IP查询

当想要知道一个IP地址，对应着什么城市地区的话，可以用在线网站去查询IP。

举例：

[IP地址查询](#)

查询结果：

您的IP地址是：[122.193.30.34] 来自：江苏省苏州市 联通



www.ip138.com IP查询(搜索IP地址的地理位置)

您的IP地址是：[122.193.30.34] 来自：江苏省苏州市 联通

在下面输入框中输入您要查询的IP地址或者域名，点击查询按钮即可查询该IP归属地。

IP地址或者域名	122.193.30.34	查询
IP查询接口 劫持检测 公共DNS ip计算器		

另外还有一个更好的查询IP的网站，支持查询自己翻墙的IP地址

[显示查询自己的IP地址](#)

<http://ip111.cn>

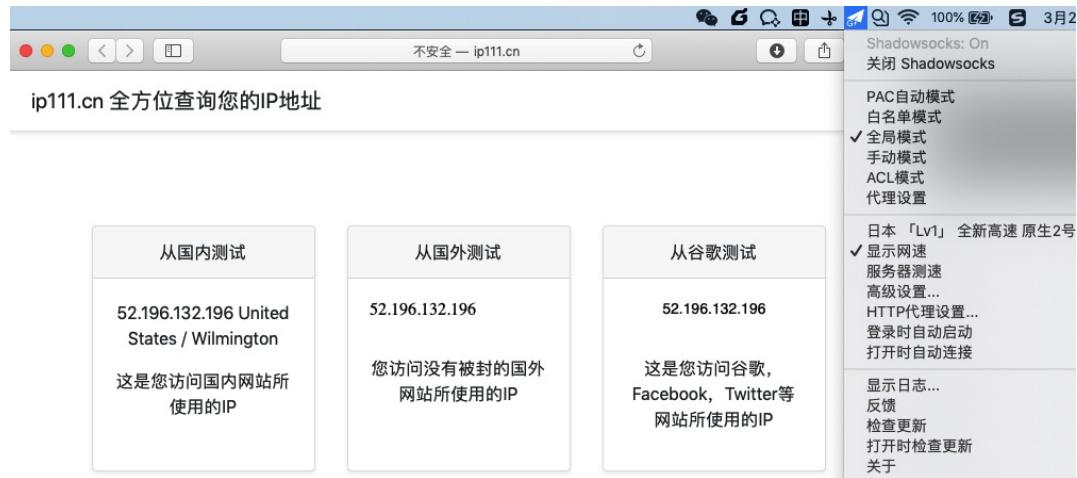
- 查询结果举例：

- 没开代理



从国内测试 122.193.30.34 China / Nanjing 这是您访问国内网站所使用的IP	从国外测试 122.193.30.34 您访问没有被封的国外网站所使用的IP	从谷歌测试 52.196.132.196 这是您访问谷歌，Facebook, Twitter等网站所使用的IP
---	--	---

- 开了全局代理



crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09
10:14:59

Windows

此处整理Windows系统中的各种知识和经验心得。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09
10:14:59

Mac

此处整理Mac系统中各种通用知识和经验心得。

键盘

常用按键

Mac按键	键盘上标识	缩写	对应Win中的
Command键	⌘	Cmd	Windows 键
Option键	⌥		Alt
Caps Lock键	⇪ 中/英		
Shift键	⇧		
Control键	⌃	Ctrl	
Fn键	fn		
delete键	☒	del	

常用快捷键

- **Command+Shift+N**

- Finder : 新建空文件夹



- **Command+O:** 打开所选项，或打开一个对话框以选择要打开的文件

- 举例：

- Finder : (先选中某文件后) 打开文件
 - = 鼠标双击某文件

- **Command+J**

- Finder : (先选中某文件后) 打开文件
 - = 鼠标双击某文件

- **Command+↑**

- (从当前目录) 回到上一级目录

快捷键

附上别人整理，相对比较全的，最常用的Mac快捷键：



注：其中有个笔误：

- Command+Shift+N 才是‘打开文件’而不是 Command+N

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2021-01-16 21:30:25

brew

给brew加代理用于加速下载

```
export ALL_PROXY socks5://127.0.0.1:1086
```

举例：

```
export ALL_PROXY socks5://127.0.0.1:1086  
brew install cmake
```

详见：

【已解决】mac中给brew设置代理加速下载和更新

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-08-09 10:14:59

app安全限制

Mac中默认安全限制很严格。

最常见的一个现象就是：

第一级：允许任何来源解决app已损坏，打不开。您应该将它移到废纸篓

对于很多未签名的，来源不明的，但其实往往是安全的app，默认是不允许安装的。

且还会提示 app已损坏，打不开。您应该将它移到废纸篓

解决办法：

先去终端中

```
sudo spctl --master-disable
```

然后去：

设置 -> 安全与隐私 -> 通用 -> 允许从以下位置下载的app -> 任何来源



第二级：设置不要被隔离

但是个别情况，此时还无法打开app，举例：



原因： macOS 10.15+ 系统对于未签名的应用又进一步收缩了权限

解决办法=规避办法=绕开办法：

```
sudo xattr -rd com.apple.quarantine /Applications/YourApplicationName.app
```

举例：

```
sudo xattr -rd com.apple.quarantine /Applications/SecureCRT\ 2.app
```

详见：

[【已解决】Mac的Catalina中已spctl操作过但还是：app已损坏，打不开。您应该将它移到废纸篓](#)

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新：2020-08-09 10:14:59

Linux

详见独立教程：

- 【整理Book】Linux使用和开发心得

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-08-09
10:14:59

附录

下面列出相关参考资料。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09
10:14:59

参考资料

- [【已解决】mac中给brew设置代理加速下载和更新](#)
- [【整理Book】Python心得：包管理器](#)
- [【已解决】Mac中安装和使用composer安装php的库php-html-parser](#)
- [【已解决】Mac中composer install安装php的模块卡死在Updating dependencies including require-dev](#)
-
- [【已解决】Mac的Catalina中已spctl操作过但还是：app已损坏，打不开。您应该将它移到废纸篓](#)
-
- [软件包管理系统 - 维基百科，自由的百科全书](#)
- [使用 | Yarn](#)
- [Bower — a package manager for the web](#)
- [Bower: 客户端库管理工具 -- JavaScript 标准参考教程 \(alpha\)](#)
- [CocoaPods使用 - iOS - 掘金](#)
- [CocoaPods Guides - Using CocoaPods](#)
- [Maven 构建生命周期 | 菜鸟教程](#)
- [Gradle 入门--只此一篇 - 简书](#)
- [GitHub Package Registry 发布！你依旧是那个我们所爱的 GitHub！ - 知乎](#)
- [GitHub 推出包管理服务 GitHub Package Registry - OSCHINA](#)
- [About GitHub Packages - GitHub 帮助](#)
- [mac os 下打开文件夹的快捷键 | 网络进行时](#)
- [Mac 键盘快捷键 - Apple 支持](#)
- [macOS 有哪些常用的快捷键？ - 知乎](#)
-

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-08-09 10:14:59