

Crifan的电子书的使用说明

最后更新： 20221028

- Crifan的电子书的使用说明
 - 源码
 - Crifan电子书的简要说明
 - 为何要有：Crifan的电子书
 - 为何要有：Crifan的电子书的使用说明
 - 技术类
 - 技术类通用知识
 - 推荐的工具或软件
 - 科学上网
 - 编辑器和IDE
 - 好用的编辑器和IDE
 - 画图工具软件
 - 文件对比工具
 - API接口测试
 - ssh远程连接工具
 - 代码版本管理
 - 硬件类
 - 嵌入式软件
 - 开发环境+交叉编译器
 - 嵌入式Linux+嵌入式Linux驱动
 - 具体模块或领域
 - PLC
 - 大的方向=嵌入式的不同应用领域
 - 小方面来说=关于具体的模块/硬件/协议方面
 - 上层软件
 - 上层软件的通用知识
 - 字符编码
 - 正则表达式
 - 信息安全
 - Web安全

- [二进制安全](#)
- [移动端安全](#)
- [设备安全](#)
- 自动化
 - [把工作自动化的工具](#)
 - [自动化测试](#)
- [PC桌面端软件](#)
- [苹果系统](#)
- [网页Web领域开发](#)
- [移动端APP开发](#)
- [爬虫](#)
- [通信领域](#)
- [上层软件的其他领域](#)
- [Python语言](#)
- [我Crifan的技术库](#)
- [非技术类](#)
 - [电脑计算机使用类](#)
 - [工作类](#)
 - [领域相关知识](#)
 - [生活类](#)
 - [娱乐类](#)

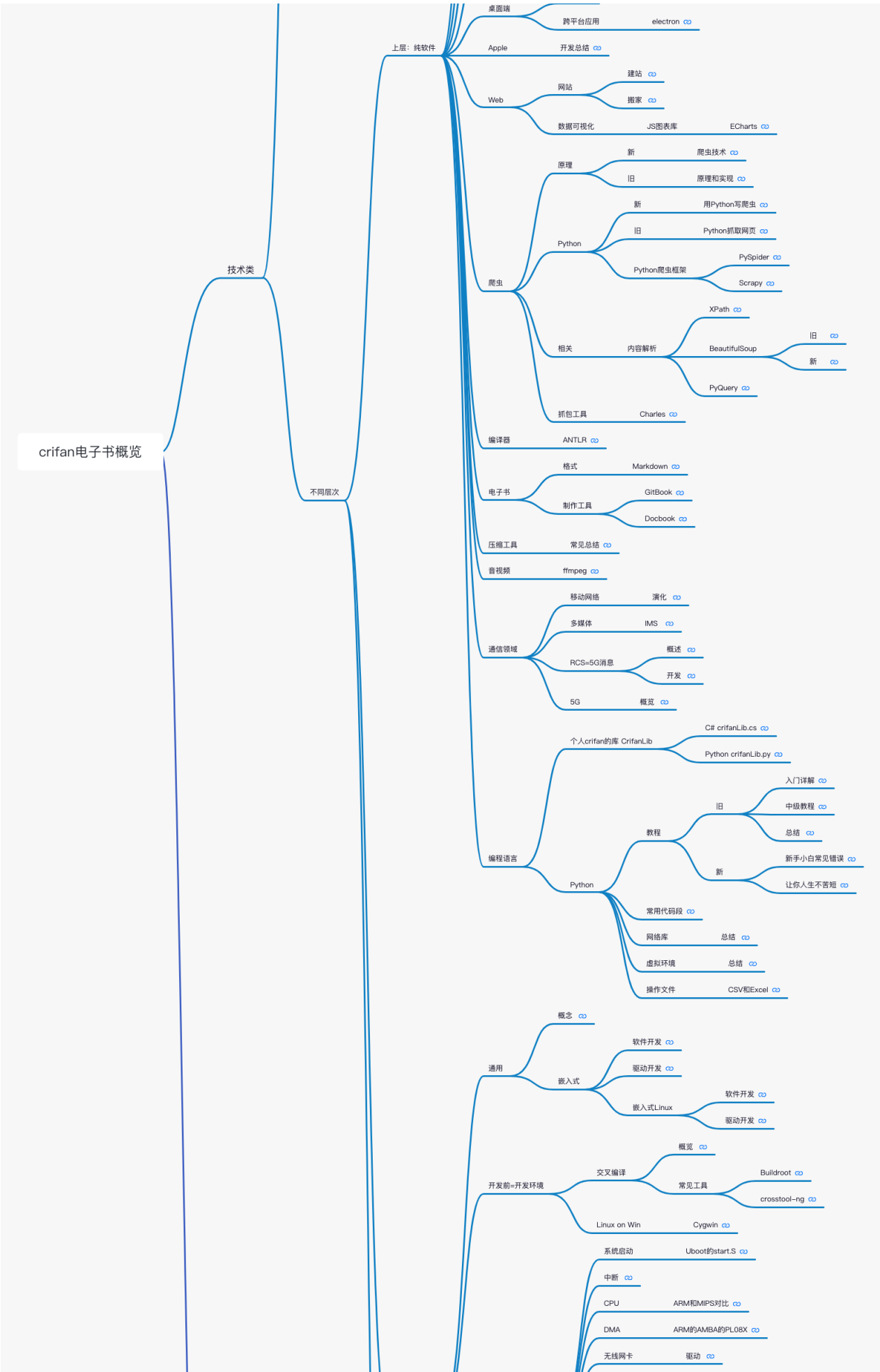
源码

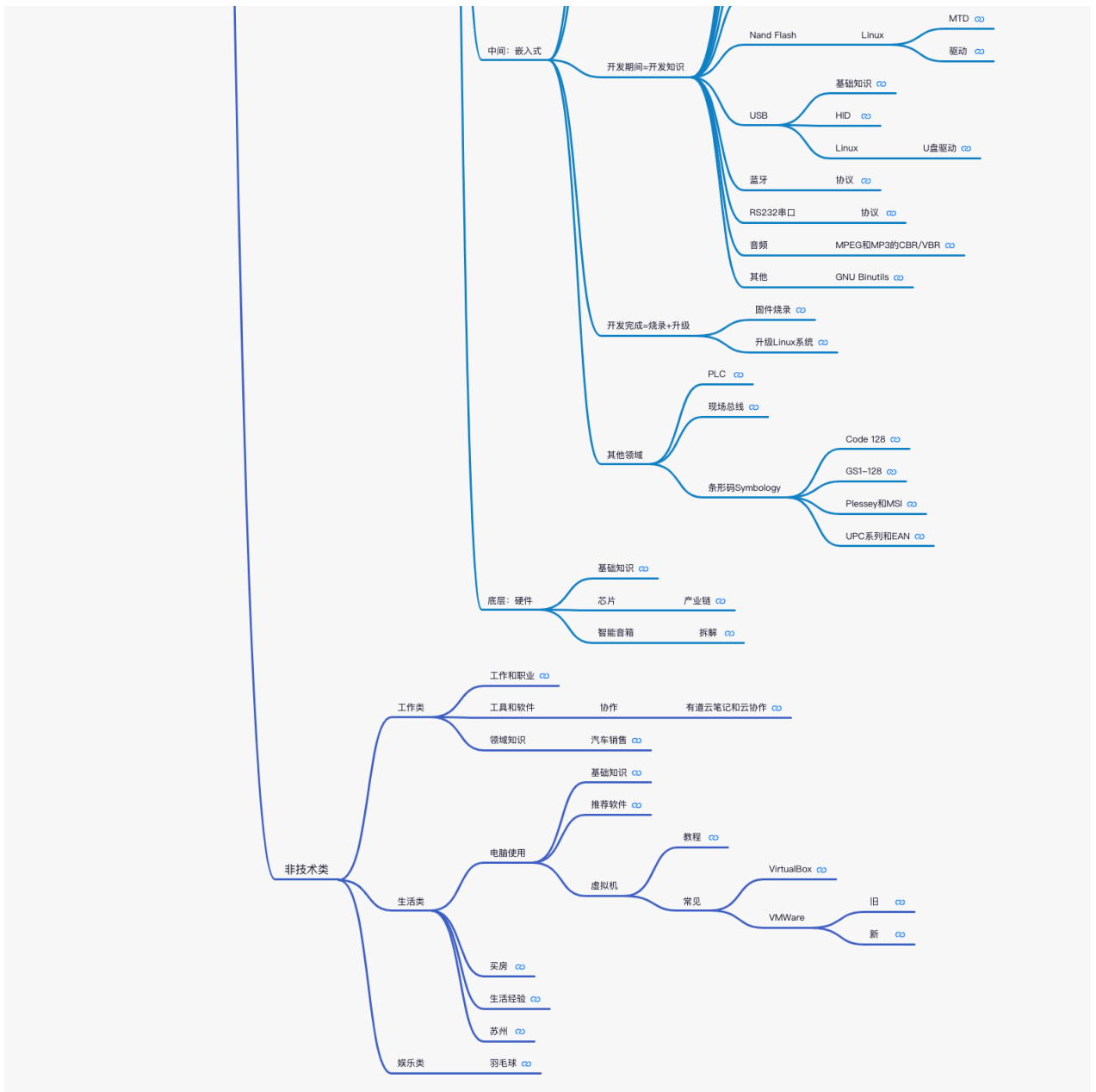
[crifan/crifan_ebook_readme](#): Crifan的电子书的使用说明

Crifan电子书的简要说明

关于[Crifan](#)的电子书，无需细节只关注重点，则只需看此概览：

- 脑图
 - 在线浏览
 - [crifan电子书概览](#)
 - 图片
 -





- 文字版

- 技术类

- 通用

- 提高效率
 - 电脑通用知识
 - 编程和开发
 - 学习方法和思路
 - 风格
 - 规范
 - 通用概念
 - 基础知识
 - 通用知识
 - 推荐工具

- [科学上网](#)
- 编辑器和IDE
 - [总结](#)
 - 好用工具
 - [Notepad++](#)
 - [VSCode](#)
 - [PyCharm](#)
- API
 - [Postman](#)
- 远程连接
 - [SecureCRT](#)
- [画图工具](#)
- [文件对比](#)
- 不同层次
 - 上层：纯软件
 - 通用
 - [HTTP](#)
 - API
 - [RESTful](#)
 - 编程语言
 - [基础知识](#)
 - [语言简介](#)
 - 数据库
 - [MySQL](#)
 - [MongoDB](#)
 - 数据格式
 - JSON
 - [旧](#)
 - [新](#)
 - 版本管理
 - [Git](#)
 - 字符编码
 - [详解](#)
 - [应用](#)
 - Python
 - [字符串和字符编码](#)
 - 正则表达式
 - [教程](#)

- 旧
 - 新
- 应用举例
- Python
 - re模块
 - 旧
 - 新
- 信息安全
 - 概览
 - Web安全
 - 渗透测试
 - 安全分析
 - 二进制安全
 - 移动安全
 - Android安全和逆向
 - Android逆向
 - Android逆向开发
 - Android逆向：动态调试
 - iOS安全和逆向
 - 旧
 - iOS安全
 - iOS安全
 - iOS安全与防护
 - iOS逆向
 - iOS逆向开发
 - 系列
 - iOS逆向开发：iPhone越狱
 - iOS逆向开发：砸壳ipa
 - iOS逆向开发：静态分析
 - iOS逆向开发：动态调试
 - iOS逆向开发：越狱插件开发
 - iOS逆向开发：iOS底层机制
 - iOS逆向开发：ObjC运行时Runtime
 - iOS逆向开发：Block匿名函数
 - iOS逆向开发：越狱检测和反越狱检测
 - iOS逆向开发：YouTube逆向
 - 相关
 - 逆向利器：IDA

- 最流行汇编语言：ARM
- Xcode内置调试器：LLDB
- 设备安全
 - 工控安全
 - 汽车安全
- 移动端
 - 安卓
 - 辅助工具
 - Xposed
 - Nox模拟器
- 自动化
 - 工具
 - Makefile
 - 测试
 - 概览
 - Web
 - Selenium
 - Playwright
 - puppeteer
 - 移动端
 - 概览
 - 通用
 - Appium
 - Android
 - uiautomator2
 - iOS
 - facebook-wda
- 桌面端
 - C#
 - 跨平台应用
 - Electron
- Apple
 - 开发总结
- Web
 - 网站
 - 建站
 - 搬家
 - 数据可视化

- JS图表库
 - [ECharts](#)
- 移动端
 - [APP开发](#)
- 爬虫
 - 原理
 - 新
 - [爬虫技术](#)
 - 旧
 - [原理和实现](#)
 - Python
 - 新
 - [用Python写爬虫](#)
 - 旧
 - [Python抓取网页](#)
 - Python爬虫框架
 - [PySpider](#)
 - [Scrapy](#)
 - 相关
 - 内容解析
 - [XPath](#)
 - BeautifulSoup
 - [旧](#)
 - [新](#)
 - [PyQuery](#)
 - 抓包工具
 - [Charles](#)
- 编译器
 - [ANTLR](#)
- 电子书
 - 格式
 - [Markdown](#)
 - 制作工具
 - [GitBook](#)
 - [Docbook](#)
- 压缩工具
 - [常见总结](#)
- 音视频

- [ffmpeg](#)
- 通信领域
 - 移动网络
 - [演化](#)
 - 多媒体
 - [IMS](#)
 - RCS=5G消息
 - [概述](#)
 - [开发](#)
 - 5G
 - [概览](#)
- 编程语言
 - 个人crifan的库 CrifanLib
 - [C# crifanLib.cs](#)
 - [Python crifanLib.py](#)
 - Python
 - 教程
 - 旧
 - [入门详解](#)
 - [中级教程](#)
 - [总结](#)
 - 新
 - [新手小白常见错误](#)
 - [让你人生不苦短](#)
 - [常用代码段](#)
 - 网络库
 - [总结](#)
 - 虚拟环境
 - [总结](#)
 - 操作文件
 - [CSV和Excel](#)
- 中间：嵌入式
 - 通用
 - [概念](#)
 - 嵌入式
 - [软件开发](#)
 - [驱动开发](#)
 - [嵌入式Linux](#)

- 软件开发
 - 驱动开发
- 开发前=开发环境
 - 交叉编译
 - 概览
 - 常见工具
 - Buildroot
 - crosstool-ng
 - Linux on Win
 - Cygwin
- 开发期间=开发知识
 - 系统启动
 - Uboot的start.S
 - 中断
 - CPU
 - ARM和MIPS对比
 - DMA
 - ARM的AMBA的PL08X
 - 无线网卡
 - 驱动
 - Nand Flash
 - Linux
 - MTD
 - 驱动
 - USB
 - 基础知识
 - HID
 - Linux
 - U盘驱动
 - 蓝牙
 - 协议
 - RS232串口
 - 协议
 - 音频
 - MPEG和MP3的CBR/VBR
 - 其他
 - GNU Binutils
- 开发完成=烧录+升级

- 固件烧录
 - 升级Linux系统
 - 其他领域
 - PLC
 - 现场总线
 - 条形码Symbology
 - Code 128
 - GS1-128
 - Plessey和MSI
 - UPC系列和EAN
 - 底层：硬件
 - 基础知识
 - 芯片
 - 产业链
 - 智能音箱
 - 拆解
- 非技术
 - 工作类
 - 工作和职业
 - 工具和软件
 - 协作
 - 有道云笔记和云协作
 - 领域知识
 - 汽车销售
 - 生活类
 - 电脑使用
 - 基础知识
 - 推荐软件
 - 虚拟机
 - 教程
 - 常见
 - VirtualBox
 - VMWare
 - 旧
 - 新
 - 买房
 - 生活经验
 - 苏州

- 娱乐类
 - 羽毛球

为何要有：Crifan的电子书

我 Crifan Li 之前折腾过很多方面的内容，包括技术类的和非技术类的。

关于技术类中又分很多种，其中都放到了我的网站 crifan.com 上的[工作和技术](#)，目前已有**7000多**的技术帖，但多数内容都是零散的某个细节知识点和某个具体问题的解决过程和结果。

当我在某个技术领域有了一定的心得和经验后，就会抽空整理成相对成系统的内容，用([Docbook](#)、[Gitbook](#)等)工具去制作出电子书，免费发布出来供别人参考。

目前已经整理出**150多**本成系列的电子书了，接下来详细介绍：

另：手上还有**几十本**电子书的内容已基本完成，待之后能抽出时间再尽快整理发布出来。

为何要有：Crifan的电子书的使用说明

我写的电子书涵盖范围面较广，需要专门整理后的分门别类的介绍，才有利于他人快速定位到自己所需要参考的内容。

目前所写电子书的总的概览：

- 内容上：主要分两类
 - 一方面：技术类
 - 计算机相关的，技术类，开发类
 - 另一方面：非技术类
 - 生活类的，学习类的，娱乐类等
- 制作工具上：也分两类
 - 一类是（最早）用 Docbook 制作的
 - 主入口：<https://www.crifan.com/files/doc/docbook/>
 - 另一类（后来，现在）是用 Gitbook 制作的
 - 弄了2份一样的，分别发布到
 - 自己的 crifan.com 的网站上
 - 主入口：<https://book.crifan.com/books/>
 - 和 [GitHub Pages](#) 上
 - 主入口：<https://crifan.github.io>
 - 对应源码：<https://github.com/crifan/crifan.github.io>

- gitbook 电子书的源码
 - 全部源码都已公开放在 github 上了，详见：<https://github.com/crifan>
 - 如何利用gitbook的源码去创建自己的电子书，可以参考：
https://github.com/crifan/gitbook_template

下面详细介绍各个电子书的具体分类和内容简介：

技术类

从大的概念上来说，计算机的技术可以大致分为：

- **上层：纯软件**
 - 涉及到在PC（Windows/Mac/Linux）上，用各种编辑器/IDE，去编写各种代码，实现各种工具/软件/脚本等内容
 - 包括PC桌面端软件，Web领域网页开发，移动端APP开发等等
- **中间：（软硬件结合的）嵌入式**
 - 涉及到在某某开发板上，写嵌入式代码，驱动硬件工作
 - 包括给别人的芯片或开发板，去写某BSP，某个模块（蓝牙，USB，FM，Nand Flash等）的驱动
- **底层：纯硬件**
 - 涉及PCB Layout，设计和生产开发版
 - 涉及到各种芯片的涉及，包括CPU，内存，显示屏等，涉及到流水线，代工厂等等内容

自己主要涉及的领域是：

中间的嵌入式 和 上层的纯软件

技术类通用知识

不论哪方面的技术，都有一些通用的学习方面，逻辑概念，总结如下：

- 关于如何学习和利用网络资料，以及一些技术方面的心得体会和总结：
 - [学习方法思路及技术心得总结](#)
- 关于如何利用工具和脚本提供工作效率的：
 - [如何提高工作效率](#)
- 关于要有良好的编程习惯和逻辑，才能写出高质量代码：
 - [编程习惯和代码风格](#)
 - [计算机语言编程规范](#)
- 对于不同领域的技术，都有一些通用的概念需要了解，才能更好的开发：
 - [计算机编程通用逻辑知识概念](#)

- 以及都有一些通知的基础知识：
 - [软件开发基础知识](#)
 - [软件技术开发通用知识](#)
- 对于计算机，或者说电脑，各种不同类型系统都有一些通用的基础知识：
 - [计算机电脑知识总结](#)

推荐的工具或软件

做各种领域的技术开发，甚至非技术方面的工作，都会涉及到一些好用的工具和利器，现整理如下：

科学上网

想要用外网，比如用google等搜索技术资料，可以使用shadowsocks：

- [科学上网相关知识总结](#)

编辑器和IDE

关于编辑器和IDE的概览和总结：

[编辑器和IDE总结](#)

好用的编辑器和IDE

之前Windows下觉得最好用的编辑器Notepad++：

- [【crifan推荐】轻量级文本编辑器，Notepad最佳替代品：Notepad++](#)

现在觉得更好用的、最好用的编辑器（甚至可称之为IDE）：VSCode：

- [史上最好用的编辑器：VSCode](#)

Python开发中最专业的IDE：

- [最智能的Python的IDE：PyCharm](#)

画图工具软件

涉及到画路程图、结构图、架构图等，可以参考：

[好用的画图工具](#)

去选择合适的画图工具。

文件对比工具

如果需要对比文件、文件夹中的内容，可以参考：

[文件内容对比工具总结](#)

API接口测试

后端人员，移动端开发人员，测试人员等涉及调试api接口时，可以用Postman：

- [API开发利器：Postman](#)

ssh远程连接工具

通过SSH连接远程服务器的工具方面，常用SecureCRT：

- [【crifan推荐】支持多种协议的串口开发工具：SecureCRT](#)

代码版本管理

当前最流行的是 Git：

- [最流行的版本管理系统：Git](#)

硬件类

折腾嵌入式期间，其实也想去了解硬件方面的知识，只不过没有深入。

只整理了点和硬件相关的皮毛：

- [硬件电路基础知识](#)

之前拆过智能音箱，然后总结了相关各种细节：

- [智能音箱拆解总结](#)

且在拆解期间又去整理了芯片和产业链的相关知识：

- [芯片产业链总结](#)

嵌入式软件

下面是之前折腾的嵌入式开发相关的内容：

嵌入式领域（包括很多硬件反面）都有些通用的基础知识和概念：

- [硬件和嵌入式通用逻辑知识概念](#)

关于CPU方面的：

- [ARM与MIPS的详细对比](#)

和技术和概念相关的，比如：

- [【详解】中断相关的知识](#)

开发环境+交叉编译器

在折腾嵌入式相关开发前，往往要先去搭建开发环境，其中就要先去搞懂：交叉编译

先去搞清楚什么是交叉编译：

- [交叉编译详解](#)

然后再去了解相关的GNU方面的工具：

- [GNU Binutils详解](#)

然后再去用具体的某个工具，去搭建交叉编译环境（和相关的工具链、rootfs等内容）：

- [Buildroot详解](#)
- [crosstool-ng详解](#)

关于开发环境的：

在Windows上折腾Linux方面的开发，用Cygwin：

- [Cygwin详解](#)

嵌入式Linux+嵌入式Linux驱动

如果对于嵌入式软件的整体概念不了解，可以去看：

- [嵌入式软件开发](#)

如果想要了解嵌入式软件中关于驱动开发的事情，可以去看：

- [嵌入式驱动开发](#)

而关于嵌入式Linux方面的软件和软件中的驱动，可以去看：

- [嵌入式Linux软件开发](#)
- [嵌入式Linux驱动开发](#)

关于嵌入式Linux开发期间的，有：

- [【详解】嵌入式开发中固件的烧录方式](#)
- [在Linux运行期间升级Linux系统（Uboot+kernel+Rootfs）](#)

而关于典型的Uboot+Kernel+Rootfs的嵌入式Linux中的Uboot：

其中的最开始的启动代码相关的Start.S的汇编代码，实现了启动硬件的功能。

而关于启动的逻辑的具体分析：

- [Uboot中start.S源码的指令级的详尽解析](#)

而关于具体的某个模块/功能方面的驱动有：

关于DMA的驱动：

- [详解ARM的AMBA设备中的DMA设备PL08X的Linux驱动](#)

关于无线网卡的驱动：

- [如何在Linux下写无线网卡的驱动](#)

关于Nand Flash方面的驱动和介绍：

- [【详解】如何编写Linux下Nand Flash驱动](#)

而其中Linux中关于Nand Flash的是MTD层，MTD中关于如何识别Nand Flash和启动过程的解析是：

- [Linux MTD下获取Nand flash各个参数的过程的详细解析](#)

而关于USB方面的协议介绍和相关驱动开发总结是：

- [USB基础知识概论](#)
- [如何实现Linux下的U盘（USB Mass Storage）驱动](#)
- [USB HID Learning Record](#)

具体模块或领域

PLC

此处暂时把PLC归为嵌入式大类。

关于PLC的知识的总结：

[工控自动化技术：PLC](#)

大的方向=嵌入式的不同应用领域

- [现场总线Fieldbus简析](#)

条形码Symbology相关的，各种不同类型的条形码的总结：

- [Code 128 Symbology Introduction](#)
- [GS1-128条形码和相关的AI及FNC1的详解](#)
- [Plessey & MSI Symbology Introduction](#)
- [UPC/UPC-A/UPC-E & EAN Barcode Symbology](#)

小方面来说=关于具体的模块/硬件/协议方面

关于蓝牙的：

- [蓝牙协议详解](#)

关于串口/RS232的：

- [RS232串口协议详解](#)

关于音频领域的：

关于MPEG和MP3的知识：

- [MPEG简介 + 如何计算CBR和VBR的MP3的播放时间](#)

上层软件

对于上层纯软件方面，也有一些总结：

上层软件的通用知识

关于后台开发和设计接口，移动端调用后台接口，测试人员测试接口和网页等相关的：

和HTTP方面的知识：

- [HTTP知识总结](#)

以及后台人员设计RESTful的API接口需要了解的：

- [HTTP后台端：RESTful API接口设计](#)

各种计算机语言通用的的方面的知识总结：

- [计算机编程语言基础知识](#)

- [各种计算机语言简介和总结](#)

在涉及到前台和后台数据交互，常用数据格式是JSON：

- 旧教程
 - [JSON详解](#)
- 新教程
 - [主流数据格式：JSON](#)

字符编码

而用不同的编辑器或IDE，以及处理文件相关的内容时，往往会涉及到文件的字符编码，可参考：

- [字符编码详解](#)

而上面的教程内容太多太杂，如果只是想要简单的使用编码方面的知识，可以直接去看：

- [字符编码应用](#)

正则表达式

不同的语言和工具中，都支持用正则表达式去实现复杂的规则去提取想要的信息。可以参考：

- 之前用docbook写的旧教程：
 - [正则表达式学习心得](#)
- 后来用gitbook写的新教程
 - [应用广泛的超强搜索：正则表达式](#)
 - 以及实际的使用案例：
 - [正则表达式应用举例](#)

信息安全

需要先对信息安全领域有个总体的了解，可参考我自己边学边整理出的：

- [信息安全概览](#)

然后再去分别了解不同领域内的安全技术：

Web安全

其中侧总基于浏览器的、Web相关的安全是：

- [防止被黑客攻击：Web安全](#)

其中Web安全包含几个主要方面：

- 渗透测试
 - [潜入你的网络：渗透测试](#)
- 模糊测试
- 安全(日志)分析
 - [掌握黑客的行踪：安全分析](#)

二进制安全

二进制安全，也称 PWN，主要侧重点是操作系统的底层机制：

- [探究底层机制：二进制安全](#)

其中对于Windows系统，就是常说的Windows的漏洞和挖掘等方向。

移动端安全

爬取安卓app的数据期间，折腾过如何破解安卓，后整理心得如下：

- [安卓应用的安全和破解](#)

以及辅助工具：

- [强大的安卓破解辅助工具：XPosed框架](#)
- [好用的安卓模拟器：夜神Nox](#)

另外，iOS设备的安全，也边学边整理出教程：

- [防止iPhone被黑：iOS安全](#)

后来又整理出iOS和Android的安全和逆向的成套教程：

- Android安全和逆向
 - Android逆向
 - [Android逆向开发](#)
 - [Android逆向：动态调试](#)
- iOS安全和逆向
 - iOS安全
 - [iOS安全与防护](#)
 - iOS逆向
 - [iOS逆向开发](#)
 - 系列
 - [iOS逆向开发：iPhone越狱](#)
 - [iOS逆向开发：砸壳ipa](#)

- [iOS逆向开发：静态分析](#)
- [iOS逆向开发：动态调试](#)
- [iOS逆向开发：越狱插件开发](#)
- [iOS逆向开发：iOS底层机制](#)
 - [iOS逆向开发：ObjC运行时Runtime](#)
 - [iOS逆向开发：Block匿名函数](#)
- [iOS逆向开发：越狱检测和反越狱检测](#)
- [iOS逆向开发：YouTube逆向](#)
- 相关
 - [逆向利器：IDA](#)
 - [最流行汇编语言：ARM](#)
 - [Xcode内置调试器：LLDB](#)

设备安全

对于 设备安全 ~ 物联网安全，包含几个子领域=不同侧重点：

- 工控领域
 - [工控安全概览](#)
- 汽车领域
 - [守护你的座驾：汽车安全](#)

自动化

把工作自动化的工具

不论是嵌入式还是上层软件，都可以用一些工具，从手动变自动，去提高做事效率，比如Makefile：

- [自动化利器：Makefile](#)

自动化测试

对于自动化测试，先看概览以了解有哪些子领域：

- [自动化测试概览](#)

然后再去了解不同子领域：

- Web端
 - Web领域的自动化测试，也叫Web网页自动化，往往也属于爬虫领域，常用工具有
 - Selenium
 - [Selenium知识总结](#)
 - Playwright

- [跨平台Web自动化神器：Playwright](#)
 - puppeteer
 - [Web前端自动化利器：puppeteer](#)
- 移动端
 - 先了解概况
 - [移动端自动化测试概览](#)
 - 再看具体的不同方向的工具
 - 通用=跨平台
 - [主流跨平台自动化框架：Appium](#)
 - Android端
 - 安卓端的话，可以用 uiautomator2
 - [安卓自动化测试利器：uiautomator2](#)
 - iOS端
 - 关于iOS端的自动化测试，可以考虑用 facebook-wda：
 - [iOS自动化测试利器：facebook-wda](#)

PC桌面端软件

在写Windows平台的桌面端软件时，可以使用 C#：

- [C#学习心得](#)

如果想要开发跨平台的桌面端应用，可以考虑选用 Electron：

- [跨平台桌面应用框架：Electron](#)

苹果系统

苹果的 Mac 、 iOS 等相关开发总结：

[苹果相关开发总结](#)

网页Web领域开发

关于如何搭建网站，以及如何给网站搬家的话，可以参考：

- [建设网站详细教程](#)
- [网站搬家详解](#)

在数据可视化方面，有很多JS图形库，图标库，其中主流的有：

- [流行的JS图表库：ECharts](#)

移动端APP开发

在涉及到移动端开发时，可以参考：

- [移动端APP开发总结](#)

爬虫

折腾网络爬虫（和模拟登录）领域的话，可以先去了解宏观逻辑和概念：

- [爬取你要的数据：爬虫技术](#)

再去选择合适的语言去写爬虫，比如目前最适合的是Python：

- [如何用Python写爬虫](#)

然后才涉及到基于某语言（Python）去具体选择什么框架（PySpider），以提高爬虫效率：

- [Python爬虫框架：PySpider](#)
- [主流Python爬虫框架：Scrapy](#)

在爬虫领域内涉及到提取内容时，往往才会涉及到XPath：

- [XPath知识总结](#)

如果想要抓包app时可以用：

- [app抓包利器：Charles](#)

另外，之前也写过相关的爬虫方面的资料：

- [详解抓取网站，模拟登陆，抓取动态网页的原理和实现（Python，C#等）](#)
- [Python专题教程：抓取网站，模拟登陆，抓取动态网页](#)

通信领域

在移动通信网络中，总体网络从1G到5G的演化，可以参考：

- [移动网络演化史](#)

期间出现了相关技术：

- (RCS底层所依赖的技术)IMS
 - [多媒体业务核心子系统：IMS](#)
- RCS

- 概览
 - [5G消息RCS技术总结](#)
- 开发
 - [RCS技术开发总结](#)

而最新的5G技术，总体概览如下：

- [5G技术概述](#)

上层软件的其他领域

比如想要开发自己特定领域内的语言的解析器，即实现自己的编译器，可以使用ANTLR：

- [ANTLR教程](#)

想要和我一样去制作出复杂的电子书，则可以：

先学习通用的语法格式Markdown：

- [最流行的文档格式：Markdown](#)

再去使用相对简单好用的工具GitBook去制作电子书：

- [电子书制作利器：GitBook](#)
 - 如果想要更简单易用更省心，可以基于我的[Gitbook的模板](#)去制作，生成的效果是[Gitbook演示](#)

也可以用更加强大但非常复杂的，技术类书籍专业制作工具Docbook：

- [Docbook开发手记](#)

在数据库方面，主流的文档数据库有：

- [主流关系数据库：MySQL](#)
- [主流文档型数据库：MongoDB](#)

不同系统中都会涉及到文件和代码等压缩，可以参考：

- [常见压缩工具总结](#)

如果涉及到音视频的处理，可以用强大的 `ffmpeg`：

- [强大的音视频处理工具：ffmpeg](#)

Python语言

折腾折腾Python期间，单独整理了一系列的内容，供参考：

- 旧的
 - 用于入门级的介绍的：
 - [python初级教程：入门详解](#)
 - 关于整体的Python的心得和总结：
 - [python中级教程：开发总结](#)
 - [Python语言总结](#)
 - 关于某个特定的模块的总结：
 - BeautifulSoup
 - 旧教程
 - [Python专题教程：BeautifulSoup详解](#)
 - 新教程
 - [网页解析利器：BeautifulSoup](#)
 - PyQuery
 - [HTML解析库Python版jQuery：PyQuery](#)
 - re模块
 - 旧的
 - [Python专题教程：正则表达式re模块详解](#)
 - 新的
 - [Python中正则表达式：re模块详解](#)
 - 字符编码
 - [Python专题教程：字符串和字符编码](#)
 - 网络库
 - [Python心得：http网络库](#)
 - 操作excel和CSV
 - [Python心得：操作CSV和Excel](#)
- 新的
 - 对于新手小白常见的错误和解决方案和引申思考：
 - [Python新手小白常见错误和问题](#)
 - 全新整理的
 - [让你人生不苦短的编程语言：Python](#)
 - 常用代码段：
 - [Python常用代码段](#)
 - 虚拟环境
 - [Python心得：虚拟环境](#)

我Crifan的技术库

在折腾技术方面，整理出一些，相对通用的库，整理出来，供参考：

目前最新的代码，都放到 [crifan的Github](#)上了：

其中关于自己的库函数，各种语言都有：

- <https://github.com/crifan/crifanLib>
 - 其中Python已单独整理出
 - <https://github.com/crifan/crifanLibPython>

其中关于C#和Python，分别写了专门的电子书解释如何使用：

- [详解crifan的C#库：crifanLib.cs](#)
- [详解crifan的Python库：crifanLib.py](#)

非技术类

电脑计算机使用类

和电脑/计算机使用方面的知识，比如有哪些常见的操作系统，以及如何安装软件和驱动等知识的：

- [电脑基础知识教程](#)

以及觉得很多好用的工具和软件，也进行了推荐：

- [crifan推荐软件](#)

和技术开发有点关系，但是更主要是属于电脑使用方面的，虚拟机：

在 Windows 、 Mac 等系统中，想要安装另外一个系统，可以用虚拟机软件，比如 VMWare 或 VirtualBox：

- [虚拟机教程](#)
- [VirtualBox教程](#)
- VMWare
 - 旧的
 - [VMWare教程](#)
 - 新的
 - [主流虚拟机软件：VMWare](#)

工作类

而开始了工作后，需要了解的各种基本概念和常识，包括招聘和应聘，薪资待遇，股票和期权，创业和公司等：

- [工作和职业相关知识](#)

而和别人合作时，往往涉及到文件共享，资料共享，协同编辑等内容，可以使用有道云协作：

- [有道云笔记和云协作使用总结](#)

而工作后，会遇到具体的某个商业领域相关的知识，比如：

领域相关知识

汽车销售行业内的知识，包括售前和售后、整车厂和经销商。经销商和客户等相关内容：

- [汽车销售领域知识总结](#)

生活类

和买房前后需要注意哪些事项，具体买房的过程的内容：

- [买房详细教程](#)

以及日常生活期间也有些心得供参考：

- [居民生活经验总结](#)

针对苏州这个城市，也专门整理了极其详尽的各种资料，尤其是宏观规划，详见：

- [中国宜居城市：苏州](#)

娱乐类

本人常年打羽毛球，有些心得，整理如下，供喜欢羽毛球运动的参考：

- [老少皆宜的运动：羽毛球](#)