1、Permission denied 权限不足
解决的办法：
$ sudo chmod -R 777 某一目录
比如设置当前目录：$ sudo chmod -R 777 ./
切忌随便在根目录敲该命令

2、shell解析json，请先安装brew，用brew安装jq。
brew安装命令：curl -LsSf http://github.com/mxcl/homebrew/tarball/master | sudo tar xvz -C/usr/local --strip 1
jq安装命令：brew install jq
brew
上传下载推荐wget工具集，安装命令brew install wget --with-libressl
wget命令集，可直接输入wget —help查看，也可参考一下链接：
常用命令参考：https://www.cnblogs.com/lxz88/p/6278268.html
详细命令参考：https://blog.csdn.net/vivian_wanjin/article/details/81413155
注意：常用参数-o与-O是有区别的，一个小写的o一个是大写的O，大写的O会下载一次文件，如果存在则覆盖，小写的o是额外把下载的文件当做记录，再次写入到指定目录。表现形式通常为下载两份文件。

提前填坑：若提示/usr/local must be writable或者chown: /usr/local: Operation not permitted，经查验，有效解决方案为，先卸载已安装的homebrew，即执行命令/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/uninstall)”，随后重新安装，/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
即可解决之前的问题。参考：https://blog.csdn.net/yemao_guyue/article/details/80575532

3、shell进行解压缩时，可能遇到以下几种情况：
① 解压到了当前cd的文件夹目录，此时可以指定解压的目标目录：-d ./Download/AppIcon
② mac系统会额外生成一个__MACOSX的目录，且其中包含另一份相同的解压文件，此时可以跳过mac系统特有的文件夹，具体操作则是在命令最后加上：-x __MACOSX/*
③ 重复解压同一个zip，终端会提示你选择覆盖单个、全部等等完成解压操作，这时你可以在解压命令前加上参数-o,直接覆盖
完整的解压命令，如：unzip -o ./Download/AppIcon/AppIcon.appiconset.zip -d ./Download/AppIcon -x __MACOSX/*

4、使用xcodebuild构建archive包，根据项目是否使用到cocoapods，分别为：
使用cocoapods：
xcodebuild archive -workspace 项目名称.xcworkspace -scheme 项目名称 -configuration Release -archivePath archive包存储路径
CODE_SIGN_IDENTITY="证书" PROVISIONING_PROFILE="描述文件UUID"

未使用到cocoapods：
xcodebuild archive -project 项目名称.xcodeproj -scheme 项目名称 -configuration Release -archivePath archive包存储路径 CODE_SIGN_IDENTITY=证书 PROVISIONING_PROFILE=描述文件UUID
参考链接：https://www.jianshu.com/p/f50053d50436

5、执行xcodebuild archive报错
error: tool 'xcodebuild' requires Xcode, but active developer directory
需要切换到当前Xcode路径下，执行以下命令即可
sudo xcode-select --switch /Applications/Xcode.app/Contents/Developer/

6、在xcodebuild -exportArchive之前一定要配置-exportOptionsPlist，这是xcode9之后增加进来的，必须配置的参数有teamID（企业签名不要），method（ad-hoc,app-store,enterprise,development）,provisioningProfiles(如：XFlee.FastLaneTestDemo AdHoc)
注意：provisioningProfiles是一个字典，其元素key为bundleID,值为描述文件名称
ps：该标准格式的ExportOptions.plist可以从手动打包出来的文件夹中获取，另外里边的key注意别多了空格，不容易直接肉眼识别

7、security
```
#删除证书秘钥
/usr/bin/security delete-certificate -c "iPhone Distribution: jiebing xu”
```

8、描述文件注册进资源库的路径
~/Library/MobileDevice/Provisioning/${mobileprovision_uuid}.mobileprovision

9、表单上传使用curl命令说明：
-F：表单文件
-D：data文件
文件路径前注意要用@引用，e.g. curl -F "upload=@$post_file" -F "type=ipa" "https://www.vduan.top/file/upload"

126.搭建FTP服务器，参考连接：https://blog.csdn.net/qq_33862644/article/details/80588398
先使用命令sudo su退回shell根目录，
使用root用户登录服务器命令：ssh -p 端口号 用户名@ip地址 注：端口号默认：22
首先安装vsftpd命令：yum install vsftpd -y
启动vsftpd服务：systemctl start vsftpd.service
重启服务：service vsftpd restart
添加用户：useradd -d /usr/Ftptest test1
为test1设置密码：passwd test1
强制删除用户：userdel -rf test1

使用sftp上传，为了避免输入密码的手动操作，可以先安装expect组件，安装命令如下：

```
brew install expect
```

mac客户端安装ftp命令集：
```
brew install telnet
brew install inetutils
brew link --overwrite inetutils
```
参考链接：https://www.jianshu.com/p/10c4e46c77f1
额外补充：windows终端或linux使用yum

语法注意事项：

1、赋值语法“=”前后不能有空格

2、多参数输入，使用空格分开
e.g.
```
echo "Please enter some paramters:"

read name password age sex

echo "User: $name Password: $password Age:$age Sex:$sex"
```

若在输出字符串中使用的参数能正常识别，可以使用上双引号单独引用参数，比如：
```
echo “"User:"$name"Password:"$password"Age:"$age"Sex:"$sex""
```
或者可以单独使用{}释义变量名的范围，如下
```
echo "User: ${name} Password: ${password} Age:${age} Sex:${sex}”
```

3、单引号与双引号的区别
单引号：包围变量的值时，单引号里面是什么就输出什么，即使内容中有变量和命令（命令需要反引起来）也会把它们原样输出。这种方式比较适合定义显示纯字符串的情况，即不希望解析变量、命令等的场景。
双引号：包围变量的值时，输出时会先解析里面的变量和命令，而不是把双引号中的变量名和命令原样输出。这种方式比较适合字符串中附带有变量和命令并且想将其解析后再输出的变量定义。

4、shell中定义的变量，默认为全局变量，即使是在函数内部定义的，如果想将其定义为局部变量，则可在变量前加local声明。
5、标点符号的作用：
"执行变量替换，返回字符串"
'不识别变量，直接返回输入字符串'
`执行命令`

6、jq工具语法：
基础语法：jq + 字符串 执行jq命令
e.g. 未规范格式的json字符串 | jq .

可返回格式化json字符串,所以通常jq语法后边一定先跟一个英文句号.部分情况是省略了单引号的写法。规范写法应该是 | jq '.' 或者 | jq '.msg'

echo命令直接加 | jq . 可输出规范格式且字体高亮的json格式文本
但是如果使用变量记录 | jq .msg 则无法记录样式，只能记录取到的值

-r 参数可以去除取得值的双引号，如：jq -r .msg

7、shell 读取本地文件：
e.g. json=`cat hello.text | jq .` 注：此时$json获得hello.text的文本内容
读取网络即时文件：
e.g. json=`curl 'http://118.25.21.220:18731/hello' | jq '.'`

8、拷贝粘贴命令
cp [源文件目录] [目标文件目录]
cp后可加参数，详细参数可参考https://blog.csdn.net/leon1741/article/details/54425222

9、shell的判断语句，使用[判断条件]，判断条件与中括号之间需要用空格隔开

10、当文件路径中有空格时，使用转义字符"\ ",反斜杠+空格的形式表示空格

附加：　　关于脚本语言及工具的学习使用技巧
1、脚本语言的语法比较类似，比如shell，它就像一个为其他工具集提供工作环境的一个工作区域，它有它的基础设施，也就是它的基本语法。

2、shell的方法定义，类似C语言，不是，好像就OC不一样。

3、关于shell和ruby的入参，参数都是以空格的方式区分参数，读取参数时候是以数组的方式读取，比如shell中比较特别的是第一个入参$0是和你执行shell文件有关,比如你是用命令sh test.sh的方式，那么$0的值则为test.sh,后边的参数依次为$1,$2…

4、关于shell的符号，我想说提出下边这几个符号：双引号，单引号，间隔号（就是esc下边的那个，中文环境下是波浪号），井号。下边说说他们的特点
引号：双引号里边的内容是会进行变量读取的，而单引号只会直接认为是字符串，比如假设$a的值为20，那么echo "我今年$a"。双引号的结果是我今年20，单引号的结果就是我今年$a。
间隔号：引用的是命令，返回结果是该命令的执行结果，比如查询结果，解析结果等。
井号：正常都是用作注释，但是有个特殊的用法是读取数组的count，如：${#array[@]}或${#array[*]}

5、下边介绍一些工具集使用的共通性，正常的用法都是[工具集名 [可选参数] [方法名]]，如brew install wget。注：可选参数为工具集的可选参数而非方法的参数，方法的参数使用空格隔开。可选参数通常如-P的形式，不同工具集参数可能一样，注意别用串了。