**Web端爬虫：**

1. Puppeteer（类似于selenium，是一种自动化测试工具，但功能更强大，适用于那些有许多加密参数的网站登录或爬取）

入门：<https://www.jianshu.com/p/679f07ba474b>

视频：B站搜BenAward

文档：<https://zhaoqize.github.io/puppeteer-api-zh_CN/#/>

1. requests（python的库，使用的前提是必须保证请求的参数都已经获取到，明白参数的生成规则，并且可以用python来真实模拟这些参数的生成，保证爬虫的稳定性）
2. 逆向（这里主要指js的逆向，比如在某个请求中遇到了一个加密的参数，如何得知这个参数是怎么生成的，这就需要js逆向了）

* 入门例子：B站搜av78359263
* 美团实战：<https://zhuanlan.zhihu.com/p/81773389>

**App爬虫：（思路和web一样，用到的工具会有不同）**

1. mitmproxy

它可以截取手机app的数据请求，类似于Charles等抓包工具，但是强大之处在于他可以直接对接python脚本对截取的数据包进行处理。（windows环境下使用的是其中的mitmdump）

1. Appium或Airtest

这俩是APP的自动化测试工具，编写好脚本之后，可以控制手机进行一系列的操作（Appium总出问题，而Airtest简单易上手）

Airtest实战：<https://www.cnblogs.com/xieqiankun/p/use_airtest.html>

1. mitmdump+Airtest

其实就是以上两种工具的结合，先写一个脚本来控制手机，再写一个脚本来拦截数据包，这就构成了一个APP爬虫（缺点是不稳定，优点是不用管各种加密参数，直接开爬）

四、requests（同web端）

五、逆向

这里指的是apk的逆向，通过分析apk源码，得出加密的参数的生成方式，实现稳定的爬虫

饿了么app逆向实战：<https://bbs.pediy.com/thread-255543.htm>

**反爬：**

1. ip限制，解决办法：使用ip代理

示例：<https://www.cnblogs.com/z-x-y/p/9355223.html>

1. 请求中增加加密参数：

解决办法：逆向出加密参数，先试用python模拟参数生成，再使用requests进行爬取。如果逆向不出来，可以直接使用mitmdump+Airtest直接控制手机，并且截获手机的数据包

1. 美团（h5，小程序）检测到爬虫会返回假数据，饿了么（h5）则是出现一个滑动条。
2. 美团h5请求来的数据中数字是加密的。
3. 美团app，稍微修改其中的参数就会导致请求被拒绝。
4. 美团对访问的时间可能有限制，在晚上爬取可能被封的可能性较大，两次爬取的时间间隔也应该设置为随机数
5. 美团，饿了么爬取的过程中需要用户登录，提高了爬取的成本