我们在浏览网站的时候经常会遇到各种各样的验证码，在多数情况下这些验证码会出现在登录账号的时候，也可能会出现在访问页面的过程中，严格来说，这些行为都算验证码反爬虫。

本课时我们就来介绍下验证码反爬虫的基本原理及常见的验证码和解决方案。

验证码

验证码，全称叫作 Completely Automated Public Turing test to tell Computers and Humans Apart，意思是全自动区分计算机和人类的图灵测试，取了它们关键词的首字母变成了 CAPTCHA，它是一种用来区分用户是计算机还是人的公共全自动程序。

它有什么用呢？当然很多用处，如：

- 网站注册的时候加上验证码，可以一定程度上防止恶意大批量注册。
- 网站登录的时候加上验证码，可以一定程度上防止恶意密码爆破。
- 网站在发表评论的时候加上验证码，可以在一定程度上防止恶意灌水。
- 网站在投票的时候加上验证码，可以在一定程度上防止恶意刷票。
- 网站在被频繁访问的时候或者浏览行为不正常的时候，一般可能是遇到了爬虫，可以一定程度上防止爬虫的爬取。

总的来说呢，以上的行为都可以称之为验证码反爬虫行为。使用验证码可以防止各种可以用程序模拟的行为。有了验证码，机器要想完全自动化执行就会遇到一些麻烦，当然这个麻烦的大小就取决于验证码的破解难易程度了。

验证码反爬虫

那为什么会出现验证码呢？在大多数情形下是因为网站的访问频率过高或者行为异常，或者是为了直接限制某些自动化行为。归类如下：

- 很多情况下，比如登录和注册，这些验证码几乎是必现的，它的目的就是为了限制恶意注册、恶意爆破等行为，这也算反爬的一种手段。
- 一些网站遇到访问频率过高的行为的时候，可能会直接弹出一个登录窗口，要求我们登录才能继续访问，此时的验证码就直接和登录表单绑定在一起了，这就算检测到异常之后利用强制登录的方式进行反爬。
- 一些较为常规的网站如果遇到访问频率稍高的情形的时候，会主动弹出一个验证码让用户识别并提交，验证当前访问网站的是不是真实的人，用来限制一些机器的行为，实现反爬虫。

这几种情形都能在一定程度上限制程序的一些自动化行为，因此都可以称之为反爬虫。

验证码反爬虫的原理

在模块一的时候，我们已经讲到过 Session 的基本概念了，它是存在于服务端的，用于保存当前用户的会话信息，这个信息对于验证码的机制非常重要。

服务端是可以往 Session 对象里面存一些值的，比如我们要生成一个图形验证码，比如 1234 这四个数字的图形验证码。

首先客户端要显示某个验证码，这个验证码相关的信息肯定要从服务器端来获取。比如说请求了这个生成验证码的接口，我们要生成一个图形验证码，内容为 1234，这时候服务端会将 1234 这四个数字保存到 Session 对象里面，然后把 1234 这个结果返回给客户端，或者直接把生成好的验证码图形返回也是可以的，客户端会将其呈现出来，用户就能看到验证码的内容了。

用户看到验证码之后呢，就会在表单里面输入验证码的内容，点击提交按钮的时候，这些信息就会又发送给服务器，服务器拿着提交的信息和 Session 里面保存的验证码信息后进行对比，如果一致，那就代表验证码输入正确，校验成功，然后就继续放行恢复正常状态。如果不一致，那就代表校验失败，会继续进行校验。

目前市面上大多数的验证码都是基于这个机制来实现的，归类如下：

- 对于图形验证码，服务器会把图形的内容保存到 Session，然后将验证码图返回或者客户端自行显示，等用户提交表单之后校验 Session 里验证码的值和用户提交的值。
- 对于行为验证码，服务器会做一些计算，把一些 Key、Token 等信息也储存在 Session 里面，用户首先要完成客户端的校验，如果校验成功才能提交表单，当客户端的校验完成之后，客户端会把验证之后计算产生的 Key、Token、Code 等信息发送到服务端，服务端会再做一次校验，如果服务端也校验通过了，那就算真正的通过了。
- 对于手机验证码，服务器会预先生成一个验证码的信息，然后会把这个验证码的结果还有要发送的手机号发送给短信发送服务商，让服务商下发验证码给用户，用户再把这个码提交给服务器，服务器判断 Session 里面的验证码和提交的验证码是否一致即可。

还有很多其他的验证码，其原理基本都是一致的。

常见验证码

下面我们再来看看市面上的一些常见的验证码，并简单介绍一些识别思路。

图形验证码

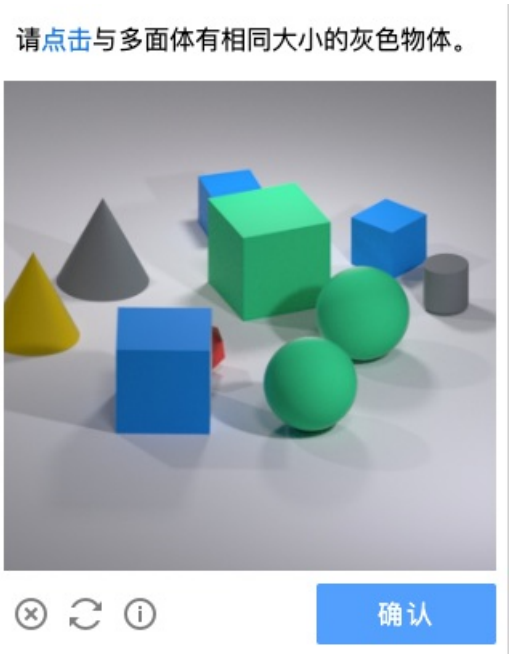
最基本的验证码就是图形验证码了，比如下图。

一般来说，识别思路有这么几种：

- 利用 OCR 识别，比如 Tesserocr 等库，或者直接调用 OCR 接口，如百度、腾讯的，识别效果相比 Tesserocr 更好。
- 打码平台，把验证码发送给打码平台，平台内实现了一些强大的识别算法或者平台背后有人来专门做识别，速度快，省心。
- 深度学习训练，这类验证码也可以使用 CNN 等深度学习模型来训练分类算法，但是如果种类繁多或者写法各异的话，其识别精度会有一些影响。

行为验证码

现在我们能见到非常多类型的行为验证码，可以说是十分流行了，比如极验、腾讯、网易盾等等都有类似的验证码服务，另外验证的方式也多种多样，如滑动、拖动、点选、逻辑判断等等，如图所示。



这里推荐的识别方案有以下几种：

- 打码平台，这里面很多验证码都是与坐标相关的，我们可以直接将验证码截图发送给打码平台，打码平台背后会有人帮我们找到对应的位置坐标，获取位置坐标之后就可以来模拟了。这时候模拟的方法有两种，一种是模拟行为，使用 Selenium 等实现，模拟完成之后通常能登录或者解锁某个 Session 封锁状态，获取有效 Cookies 即可。另一种是在 JavaScript 层级上模拟，这种难度更高，模拟完了可以直接获取验证码提交的一些 Token 值等内容。
- 深度学习，利用一些图像标注加深度学习的方法同样可以识别验证码，其实主要还是识别位置，有了位置之后同样可以模拟。

短信、扫码验证码

另外我们可能遇到一些类似短信、扫码的验证码，这种操作起来就会更加麻烦，一些解决思路如下：

- 手机号可以不用自己的，可以从某些平台来获取，平台维护了一套手机短信收发系统，填入手机号，并通过 API 获取短信验证码即可。
- 另外也可以购买一些专业的收码设备或者安装一些监听短信的软件，它会有一些机制把一些手机短信信息导出到某个接口或文本或数据库，然后再提取即可。
- 对于扫码验证的情况，如果不用自己的账号，可以把码发送到打码平台，让对方用自己的账号扫码处理，但这种情况多数需要定制，可以去跟平台沟通。另外的方案就涉及到逆向和破解相关的内容了，一般需要逆向手机 App 内的扫码和解析逻辑，

然后再模拟，这里就不再展开讲了。

基本上验证码都是类似的，其中有一些列举不全，但是基本类别都能大致归类。

以上我们就介绍了验证码反爬虫的基本原理和一些验证码识别的思路。在后面的课时我会介绍使用打码平台和深度学习的方式来识别验证码的方案。