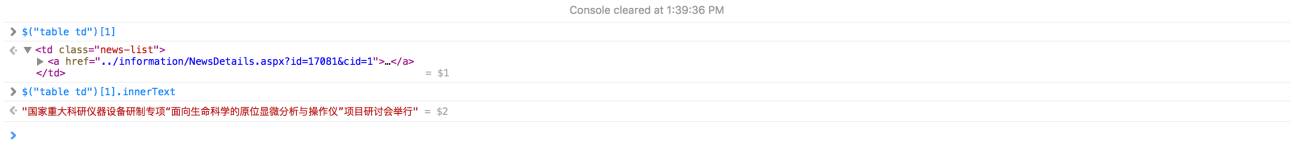
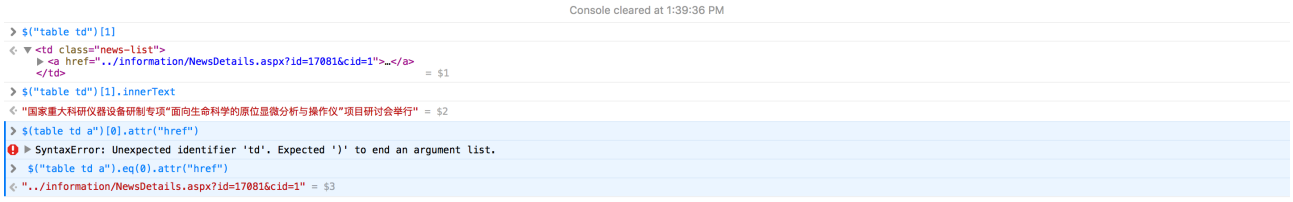
$(".s_ipt").val()
$(".s_ipt").val("南开大学")

> $("table td")[1]
< <td class="news-list">
…

</td>
> $("table td")[1].innerText
< "国家重大科研仪器设备研制专项“面向生命科学的原位显微分析与操作仪”项目研讨会举行”



> $(table td a")[0].attr("href")
< SyntaxError: Unexpected identifier 'td'. Expected ')' to end an argument list.
> $("table td a").eq(0).attr("href")
< "../information/NewsDetails.aspx?id=17081&cid=1”



get-从指定的资源请求数据
post-向指定的资源提交要被处理的数据

GET方法
查询字符串（名称/值对）是在GET请求的URL中发送的
/test/demo_form.asp?name1=value1&name2=value2

- Get请求可被缓存
- Get请求保留在浏览器历史记录中
- get请求可被收藏为书签
- get请求不应在处理敏感数据时使用
- Get请求长度有限制
- get请求只应当用于取回数据

POST方法
查询字符串（名称/值对）是在post请求的http消息主体中发送的
POST /test/demo_form.asp HTTP/1.1
Host: w3schools.com
name1=value1&name2=value2

- post请求不会被缓存
- post请求不会保留在浏览器历史记录中
- post不能被收藏为书签

post请求对数据长度没有要求

	GET	POST
后退按钮/刷新	无害	数据会被重新提交（浏览器应该告知用户数据会被重新提交）。
书签	可收藏为书签	不可收藏为书签
缓存	能被缓存	不能缓存
编码类型	application/x-www-form-urlencoded	application/x-www-form-urlencoded 或 multipart/form-data。为二进制数据使用多重编码。
历史	参数保留在浏览器历史中。	参数不会保存在浏览器历史中。
对数据长度的限制	是的。当发送数据时，GET 方法向 URL 添加数据；URL 的长度是受限制的（URL 的最大长度是 2048 个字符）。	无限制。
对数据类型的限制	只允许 ASCII 字符。	没有限制。也允许二进制数据。
安全性	与 POST 相比，GET 的安全性较差，因为所发送的数据是 URL 的一部分。 在发送密码或其他敏感信息时绝不要使用 GET ！	POST 比 GET 更安全，因为参数不会被保存在浏览器历史或 web 服务器日志中。
可见性	数据在 URL 中对所有人都是可见的。	数据不会显示在 URL 中。

方法	描述
HEAD	与 GET 相同，但只返回 HTTP 报头，不返回文档主体。
PUT	上传指定的 URI 表示。
DELETE	删除指定资源。
OPTIONS	返回服务器支持的 HTTP 方法。
CONNECT	把请求连接转换到透明的 TCP/IP 通道。

get参数通过url传递，post放在request body中
Get请求在url中传递的参数是有长度限制的，而post没有
get比post更不安全，因为参数直接暴露在url中，所以不能用来传递敏感信息
get请求只能进行url编码，而post支持多种编码方式
Get请求参数会完整保留在浏览历史记录里，而post中的参数不会被保留
对于get方式的请求，浏览器会把http header和data一并发送出去，服务器响应200（返回数据）
对于post，浏览器会发送header，服务器响应100 continue，浏览器再发送data，服务器响应200 ok（返回数据）
一般我们在浏览器输入一个网址访问网站都是get请求
http定义了与服务器交互的不同方法，其中最基本的四种，get、post、put、delete、head。其中get和head被称为安全方法，因为使用get和head的http请求不会产生什么动作。不会产生动

作意味着get和head的http请求不会再服务器上产生任何结果。但是安全方法并不是什么动作都不产生，这里的安全方法仅仅指不会修改信息
根据http规范，post可能会修改服务器上的资源的请求。比如csdn的博客，用户提交一篇文章或者一个读者提交评论是通过post请求来实现的，因为再提交文章或者提交后资源（即某个页面）不同了，或者说资源被修改了，这些便是“不安全方法”。