

---

# 个人作业1——Web前端初探

---

## 互联网数据库课程实验报告1

---

学院：网络空间安全学院

专业：密码科学与技术

学号：2112155

姓名：梁婧涵

---

### 实验要求：

1. 针对任意网页，调研其不同方式请求，至少包括get、post请求，写出或截图其请求及相应数据包
  2. 针对任意网页，使用Jquery，能够触发某一事件，写出至少三条语句，截图响应前后不同的状态
  3. 完成一个浏览器插件，功能不限，文档中写明功能及代码
- 

### 问题1 针对任意网页，调研其不同方式请求，至少包括get、post请求，写出或截图其请求及相应数据包

1、请求 Request：浏览器根据用户请求方式，将请求相关数据按照HTTP规定的格式进行封装，然后将封装好的数据发送给指定服务器。客户端根据用户地址信息将数据发送给服务器的过程

结构 请求行：请求方式、请求的地址和HTTP协议版本 请求头：消息报头，一般用来说明客户端要使用的一些俯角消息（浏览器相关数据） 空行：位于请求航和请求数据之间，是必须要有的。 请求实体：非必须（用户数据）

#### 2、GET和POST区别

##### GET

- get请求方式，请求实体数据以?形式，跟在url后面，以&隔开
- url长度限制，不能携带大量数据
- 不安全，没有请求实体部分 POST
- 适用于数据量小、安全级别不高的请求
- post请求方式，请求数据是在请求实体中进行发送，url中看不到具体请求数据
- 安全

[illegible]

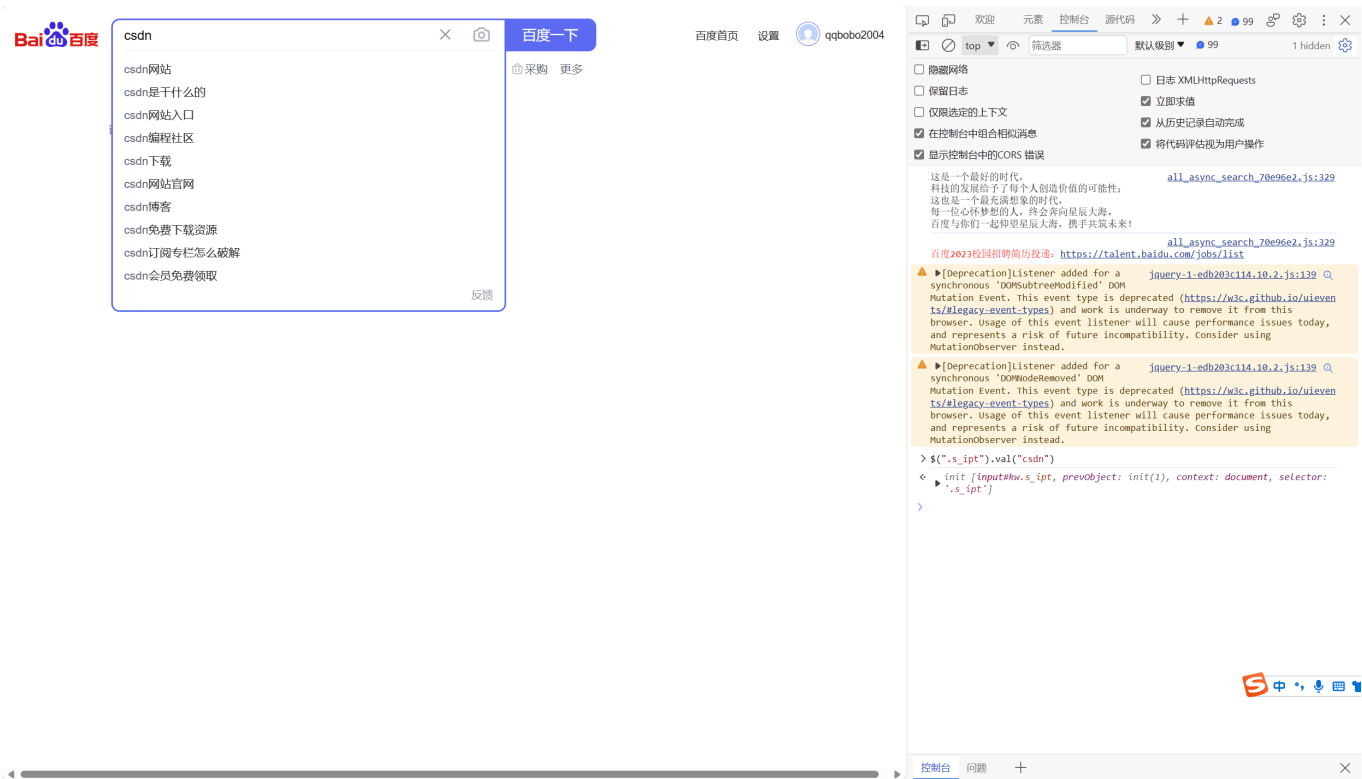
# 问题2 针对任意网页，使用Jquery，能够触发某一事件，写出至少三条语句，截图响应前后不同的状态

## 1、响应前



\$(".s\_ip").val("csdn"): 获取搜索框内的值并修改

## 响应后



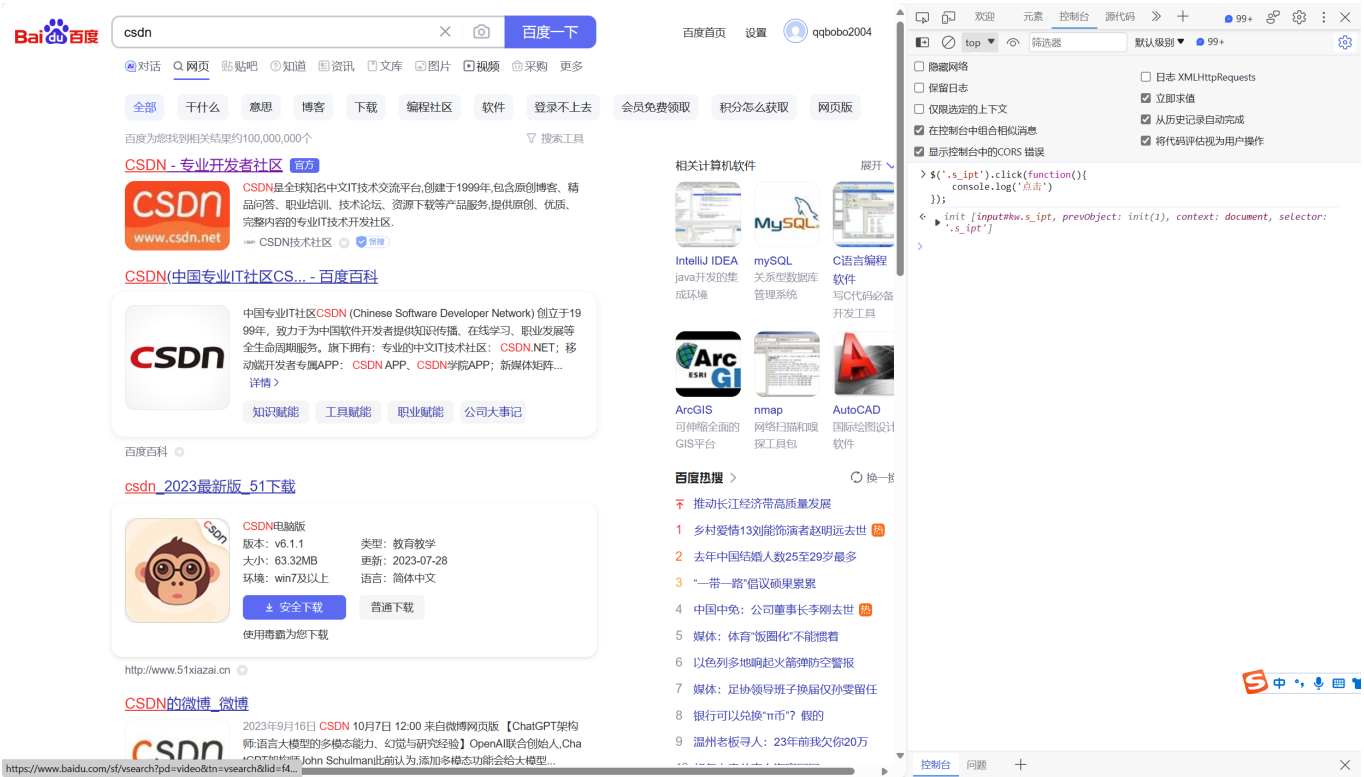
## 2、\$('.s\_ip').click(function(){

console.log('点击')

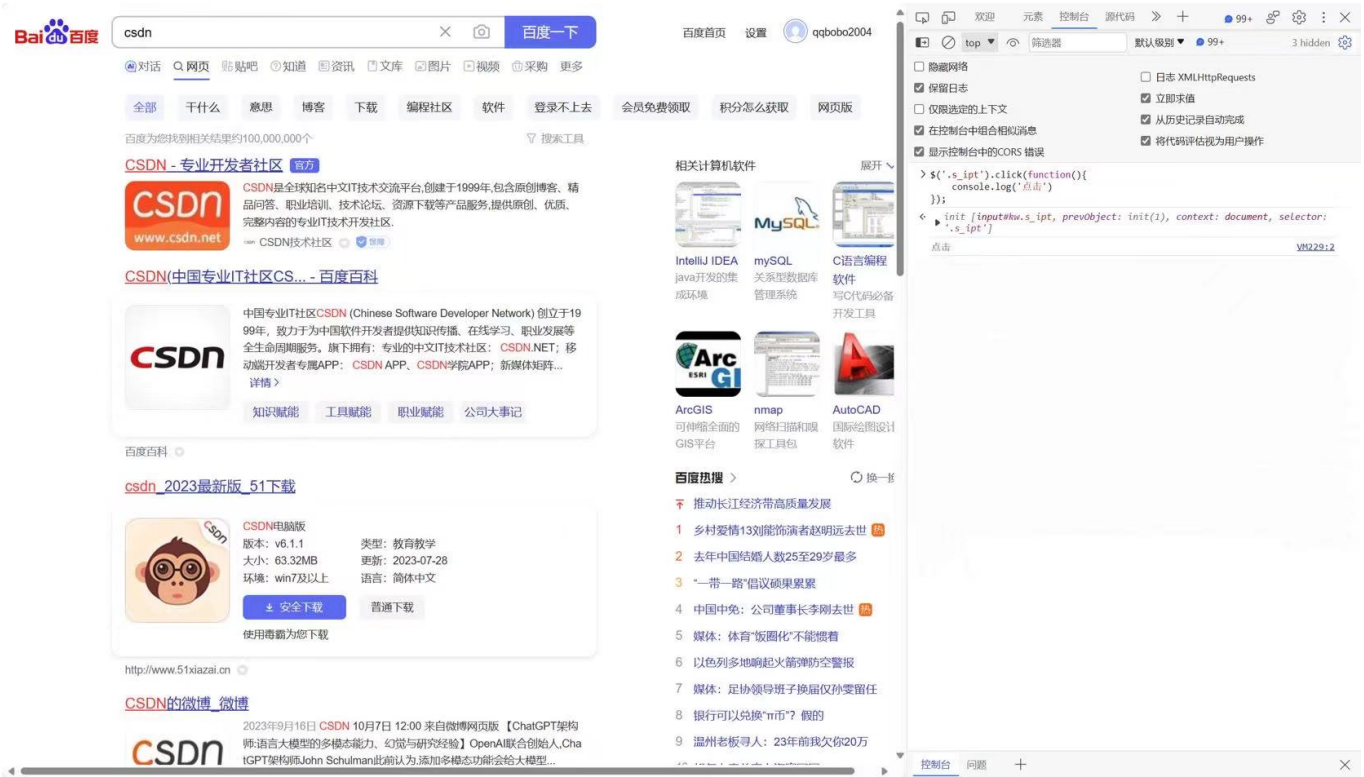
});

点击搜索框就输出点击

响应前：



响应后：



3、\$("s\_ip").click(function(){

console.log(event.offsetX, event.offsetY)



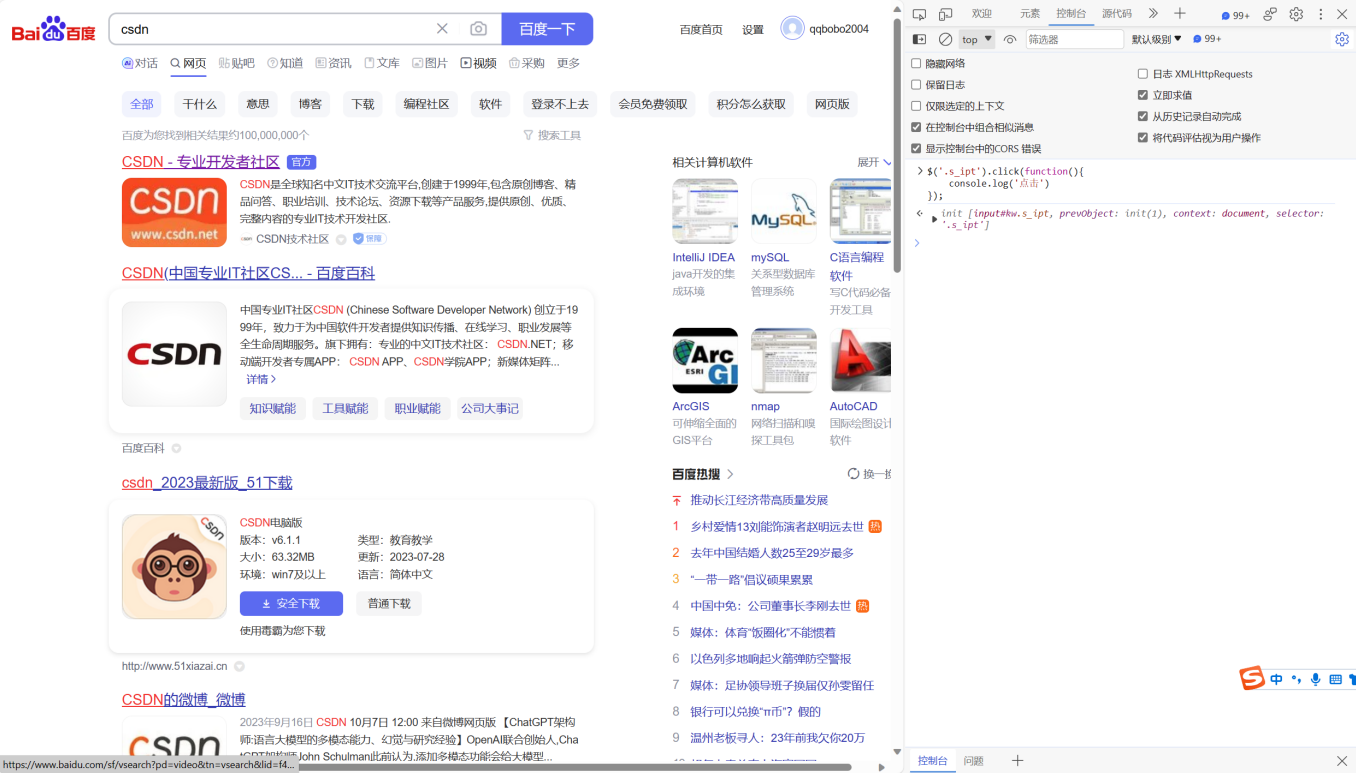
console.log(event.clientX, event.clientY)

console.log(event.pageX, event.pageY)

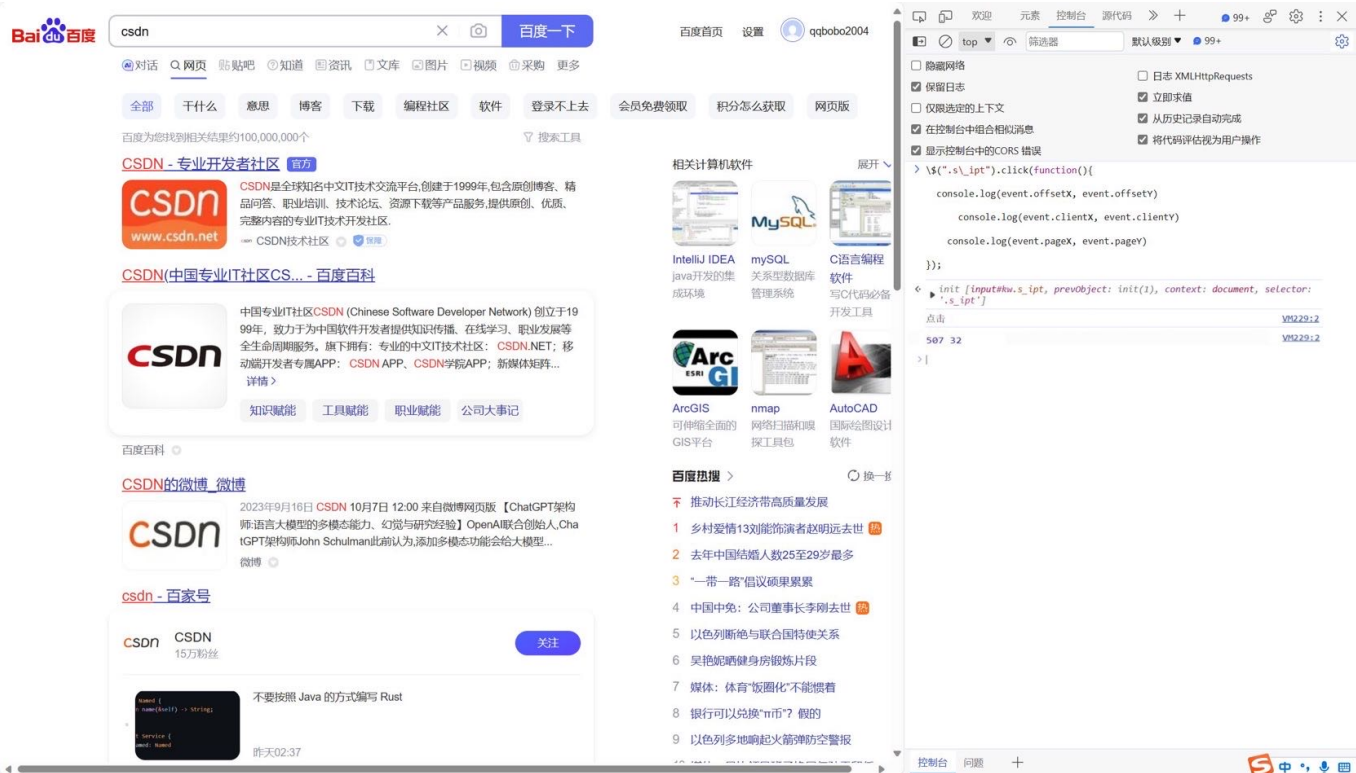
});

点击搜索框出现坐标

触发前



触发后：



## 问题3 完成一个浏览器插件，功能不限，文档中写明功能及代码

简单倒计时插件 倒计时可以是秒数，可以是时间戳，可以是日期时间（设置一种即可）

1. 支持页面内同时存在多个计时器并且不互相干扰
2. 支持多种格式，比如可以省略天、小时...等等，前面省略会自动加到后面，最多可以只有秒
3. 倒计时可以是秒数，可以是时间戳，可以是日期时间

time.js

```
/**
 * Author: CC11001100
 *
 * 简单倒计时
 *
 * 1. 支持页面内同时存在多个计时器并且不互相干扰
 * 2. 支持多种格式，比如可以省略天、小时...等等，前面省略会自动加到后面，最多可以只有秒
 * 3. 倒计时可以是秒数，可以是时间戳，可以是日期时间（设置一种即可）
 *
 * Usage:
 *
 * <div class="timer-simple-seconds" timer="3600" timestamp="1482737420000" datetime="2016-12-26 15:30:20">
 *     <span class="day">0</span>天
 *     <span class="hour">00</span>时
 *     <span class="minute">00</span>分
 *     <span class="second">00</span>秒
 * </div>
 */
/**
 * 按结构查找，依赖结构和class
 */
$(function(){

    //对所有的计时器进行处理
    var timers=$(".timer-simple-seconds");
    for(var i=0;i<timers.length;i++){
        var timer=$(timers[i]);

        if(timer.attr("timestamp")){
            //如果是时间戳，则预处理一下时间为倒计时秒数
            prepareProcessTimestamp2Timer(timer);
        }else if(timer.attr("datetime")){
            //处理时间格式为倒计时秒数
            prepareProcessDatetime2Timer(timer);
        }
        //先调用一次，避免误差
        processTimer(timer);
        setInterval(processTimer,1000,timer);
    }

    /**
     * dowhat: 这个函数将时间戳预处理成统一的倒计时描述
     *
     * 对时间做一个预处理，因为如果服务器直接返回剩余的描述的话从服务器相应到客户端虽然短到几百毫秒但总是会有偏差的，这样子不太好
     */
    }
```

```

* 所以服务器只需要设置一个时间戳表示到哪里停止就可以了
*/
function prepareProcessTimestamp2Timer(timer){
    var total=parseInt(timer.attr("timestamp"));
    total=Math.round(total/1000);
    var now=new Date().getTime()/1000;
    timer.attr("timer",total-now);
}

/**
 * 将日期时间格式转为倒计时格式
 */
function prepareProcessDatetime2Timer(timer){
    var timestamp=new Date(timer.attr("datetime")).getTime();
    timer.attr("timestamp",timestamp);
    prepareProcessTimestamp2Timer(timer);
}

/**
 * 倒计时，滴答滴答...
 * @param {Object} timer
 */
function processTimer(timer){
    var total=parseInt(timer.attr("timer"));
    var t=total;

    //倒计时不能为负
    if(total<0) return; //TODO 后续版本加上计时完毕可以回调函数

    //找到显示时间的元素
    var day=timer.find(".day");
    var hour=timer.find(".hour");
    var minute=timer.find(".minute");
    var second=timer.find(".second");

    //刷新计时器显示的值
    if(day.length){
        var d=Math.floor(t/(60*60*24));
        day.text(d);
        t-=d*(60*60*24);
    }
    if(hour.length){
        var h=Math.floor(t/(60*60));
        hour.text((h<10?"0": "")+h);
        t-=h*(60*60);
    }
    if(minute.length){
        var m=Math.floor(t/60);
        minute.text((m<10?"0": "")+m);
        t-=m*60;
    }
    if(second.length){
        second.text((t<10?"0": "")+t);
    }

    //一秒过去了...
    total--;
    timer.attr("timer",total);
}

```

```
});
```

## index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title></title>
    <style type="text/css">
      .timer-simple-seconds{
        color:#FF4040;
        font-size:20px;
      }
      .day,.hour,.minute,.second{
        font-size:50px;
      }
    </style>
  </head>
  <body>

    <span id="timer" class="timer-simple-seconds" timer="43600"><span class="day">0</span>天<span
class="hour">0</span>时<span class="minute">0</span>分<span class="second">0</span>秒
    </span>

    <script src="http://lib.sinaapp.com/js/jquery/1.9.1/jquery-1.9.1.min.js"></script>
    <script src="js/timer-simple-seconds.js"></script>
    <script type="text/javascript">

      // $("#timer").attr("timestamp",new Date("2016-12-26 15:30:20").getTime());
      // $("#timer").attr("datetime","2016-12-26 15:30:20");

    </script>
  </body>
</html>
```

## 截图展示



index.html

文件 | C:/Users/qwera/Desktop/index.html

0天12时06分38秒

DevTools is now available in Chinese!

Always match Chrome's language | Switch DevTools to Chinese | Don't show again

Elements

Console

Sources

Network

>>

top

Filter

Default levels

No Issues

```
var total=parseInt(timer.attr("timer"));
var t=total;

//倒计时不能为负
if(total<0) return; //TODO 后续版本加上计时完毕可以回调函数

//找到显示时间的元素
var day=timer.find(".day");
var hour=timer.find(".hour");
var minute=timer.find(".minute");
var second=timer.find(".second");

//刷新计时器显示的值
if(day.length){
    var d=Math.floor(t/(60*60*24));
    day.text(d);
    t--d*(60*60*24);
}
if(hour.length){
    var h=Math.floor(t/(60*60));
    hour.text((h<10?"0":"")+h);
    t--h*(60*60);
}
if(minute.length){
    var m=Math.floor(t/60);
    minute.text((m<10?"0":"")+m);
    t--m*60;
}
if(second.length){
    second.text((t<10?"0":"")+t);
}

//一秒过去了...
total--;
timer.attr("timer",total);
});
< ▶ init [document, context: document]
>
```

Console

Issues X

What's New

☐ Group by kind

☒ Include third-party cookie issues

0 0 0 0

No issues detected so far