

day13-javascript cookie

笔记本： 凡尘：二阶段

创建时间： 2017/8/26 15:31

更新时间： 2017/8/29 10:55

作者： 437389128@qq.com

通信协议

通信协议是指双方实体完成通信或服务所必须遵循的规则和约定

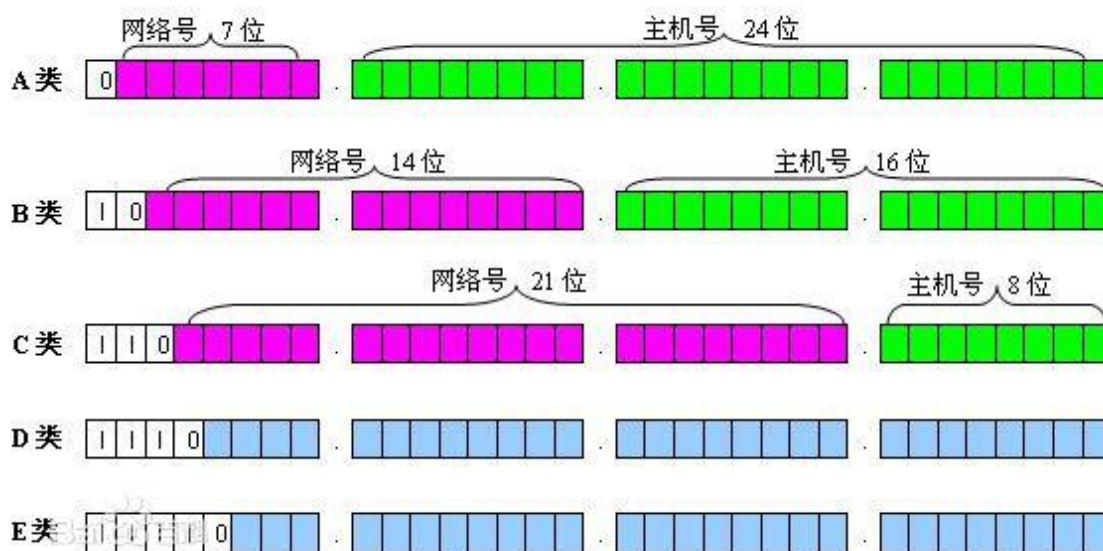
比如：地下党和总部采取密文传输 不通地区的人以普通话进行交流

IP协议 Internet Protocol

4个字节，一共32位

子网掩码 的作用？ 255.255.255.0

确认两台计算机是否处于同一网段



TCP

传输控制协议, 提供的是面向连接、可靠的字节流服务。当客户和服务器彼此交换数据前, 必须先在双方之间建立一个TCP连接, 之后才能传输数据。TCP提供超时重发, 丢弃重复数据, 检验数据, 流量控制等功能, 保证数据能从一端传到另一端。

TCP和打电话类似

双方说话之前需要一个人拨号 对方接听 然后通话 最后挂掉结束通话

建立TCP连接: 3次握手

典型举例: 魔兽世界

在别人电脑中放个大招把你干掉了，你这边瞬间漂移躲过去了。。。。游戏就没办法玩了

举例：

局域网游戏，往往都有这样的特点，当多人联机对战时，若有一人掉线

其他所有玩家进入读秒状态，那么说明玩家和玩家之间采用了TCP协议。

因为对于游戏来讲，它不允许丢失任何数据，不然有可能出现两边不一致的情况：

我这边画面一刀把你砍死了，你那边画面却吃了个大血瓶抗住了。。。。。

UDP面向数据报的协议（不可靠的协议）

UDP

用户数据报协议，是一个简单的面向数据报的运输层协议。UDP不提供可靠性，它只是把应用程序传给IP层的数据报发送出去，但是并不能保证它们能到达目的地。由于UDP在传输数据报前不用在客户和服务器之间建立一个连接，且没有超时重发等机制，故而传输速度很快

UDP类似写信，信写好了就放进邮箱 至于对方能不能收到不关心

HTTP

超文本传输协议（HTTP，HyperText Transfer Protocol）是互联网上应用最为广泛的一种网络协议。

HTTP是一个客户端和服务端请求和应答的标准（TCP）

弊端：每次请求服务器都会建立3次连接、身份确认，意味着用户每次都要登录，输入用户名和密码

这种通信最大的问题在哪？

问题就在于每次收到返回的页面，连接都断开了。

这种通讯的问题就在于，你挂了电话，老子还怎么通讯？

比如你刚刚登陆过的账号密码，如果你再想要登陆的话，还要输入一遍。。。举个简单的例子，你逛淘宝时候每看一个商品都要重新登陆一下。。。那酸爽；

COOKIE

在一次会话从开始到结束的整个过程，全程跟踪记录客户端的状态（例如：是否登录、购物车信息、是否已下载、是否已点赞、视频播放进度等等）

你把COOKIE就当成是第一次跟服务器连接后，服务器发给你的身份牌，上面就记录跟你有关的信息以后只要再跟服务器通信，必须带着这个身份牌

这样一来，关于你是谁？有没有登陆过？购物车里有什么信息？服务器当然就很容易知道了。

001、cookie使用的场景

- 1、购物车
- 2、保存用户信息
- 3、免登陆
- 4、定制页面（背景图片、所在地天气等）

002、cookie的特点

- 1、大小限制(不能超过4K)

(文件若没有大小限制，比如身份牌的重量是140斤，挂脖子能不能累死?)

- 2、每个域下cookie不能超过50个

(一般浏览器，限制大概在50条左右，你家的门禁卡里能存的下一部蓝光高清么?)

- 3、有效期（和设定时间有关），过了有效cookie会自动删除

(安全学基本理论：密码锁每次打开都需要重新输入密码，输入一次密码，以后就不再验证，就没有安全可言)

问：信用卡为什么会有过期时间?

)

- 4、cookie读取（只能访问同一个域名下的cookie）

不可跨域读取，只能由来自 写入cookie的 同一域名 的网页可进行读取。

简单的讲就是，谁写的cookie，谁才有权利读取

(身份牌是我发你的，当然只有我能读取，你媳妇儿的手机自动连接了邻居老王家的wifi，你知道这意味着什么吗?)

- 5、没有设置存储时间的cookie是临时cookie,浏览器关闭自动删除

- 6、cookie只能是字符串（文本文件）

(如果浏览器可以随意在客户端机器上生成文件，比如身份牌是个定时炸弹，安全问题将会变得非常严重)

003、写cookie(重)

```
document.cookie
```

读取所有document.cookie

004、设置cookie(重)

```
001、 document.cookie = 'cookie名=cookie值';
```

栗子: document.cookie = 'userName = pine';

```
002、设置带时间的cookie
```

```
document.cookie = 'userName = pine;expires=存储时间'
```

栗子:

```
var d = new Date();
var d.setDate(d.getDate()+3);
document.cookie = 'userName =
zhaoyahua;expires='+d.toGMTString();

02、 var d = new Date();
    d.setDate(d.getDate()+3);
    document.cookie = 'userName = zhaoyahua;expires='+d
```

005、cookie删除(重)

将存储时间设置为过去的即可

```
var d = new Date();

d.setDate(d.getDate()-1); //将时间设置为前一天

document.cookie = 'userName = zhaoyahua;expires='+d
```

006、获取cookie

007、cookie的路径path

子路径可以访问，父路径的cookie

父路径不能访问子路径的cookie

008、cookie的读、写、删除封装(重要)

练习

- 1、一周免登陆
- 2、简单购物车

json对象 (重重重)

```
1、将对象转化为json字符串  JSON.stringify(对象)
    demo:
        JSON.stringify( [1,2,3] ); // "[1,2,3]"
        JSON.stringify( {name:pine} ); // "{name:pine}"

2、将json字符串转化为json对象  JSON.parse(json字符串);

        JSON.parse( "{name:pine}" ); // {name:pine}
```

购物车应用（重）

一件商品有哪些信息？

商品编号: `gid`
商品名称: `gname`
商品价格: `gprice`
商品数量: `gamount`

购物车存储结构:

```
var cart = {
  10001:{
    gid: id,
    gname: name,
    gprice: price,
    gamount: amount
  },
  10002:{
    gid: id,
    gname: name,
    gprice: price,
    gamount: amount
  }
}
```

XSS攻击

脚本跨域攻击

在别人网站注入js代码，其他用户浏览时自动触发js执行，可以获取其他用户在网站的信息

预习

```
1、什么是正则表达式？
2、怎么创建正则表达式？
3、正则常用元字符
    ^ $
    []
    \d \w
    {n} {n,} {n,m}
    * + ?
```

\s
\b