一组

1. 浏览器从输入url到显示页面中间做了哪些工作？

浏览器会自主的根据url内容寻找服务器，经过三次握手，获取到服务器的数据流，浏览器会将获取到的数据流暂时存储在浏览器内部缓存中。然后解析数据流，html 会生成一个domtree，css会生成一个rulestree 进行页面渲染，HTML改变是重排，css样式改变是重绘，最后加载js

1. 说一说你对js执行上下文栈和作用域链的理解？

简单来说，执行上下文是基于对象，作用域链是基于函数的，我之前做的项目经常会用到构造函数，当我们new一个新对象的时候，对象函数内部的函数没执行的时候他就是一个执行上下文，同时。这个函数同属于一个作用域链，我是这么理解的。

1. 什么事闭包，闭包的作用？

闭包就是可以访问另一个函数作用域中的变量的函数，但是闭包的缺点也很明显，就是容易造成缓存泄漏问题，不过像在做轮播，时间显示等方面还必须得用闭包。

1. Jsonp的实现原理？
2. 解释js中定时器的工作？如果有，说下使用定时器的缺点？

Settimeout反复调用函数

# 二组

问题1：介绍一下闭包 和闭包常用在哪些地方？？

问题2：解释下事件代理？？

就是利用了事件冒泡，一个事件处理程序，也可以管理其某一类的所有事件

问题3：创建ajax过程？？

一创建ajax 二打开端口open 三设置头信息setrequestheader 四添加响应事件 五send发送请求

问题4：如何消除inline-block元素之间的间隙？？

设置浮动，或者把所有子元素写在一行，不能有空格

问题5：css中的link和@import的区别？？

答：link是html标签，@import是css中提供的，DOM二级中才增加了样式表对象模型，所以相比于@import，link不存在兼容问题。而且在页面加载时，link会同时被加载，而@import需要等css文件加载完才引入样式。

三组

Local storage sessions storage kookie 区别

Cookie是浏览器和服务器之间传递，local session不会传到服务器

Local是保存在本地的 没有生命周期，就是你不删除一直存在

Session仅是临时的 浏览器关了就没了

继承

闭包

数组去重

四组

1.你所知道的实现块级元素水平垂直居中的方法有几种，分别是什么？

2.js遍历的原理

3.如何实现响应式布局，原理是什么，做响应式布局的时候遇到哪些问题？

4.怎样添加、移除、替换、复制、创建和查找节点

5.请你说一下哪些情况会改变this的指向

1.你所知道的实现块级元素水平垂直居中的方法有几种，分别是什么？

答：①水平方向margin:0 auto;垂直方向用：padding设置，要知道父级盒子的宽高，而且不能浮动

②子绝父相，父级相对定位，子集绝对定位，top、left的值给50%，margin给负值

③这个也是子绝父相，top left right bottom四个属性都给0.然后margin给个auto；

应该还有其他方法，我之前做项目遇到的大概就这三种 其他的记不太清了。

2.什么情况会改变this的指向

答：函数调用的时候，call() apply() bind()

3.js遍历的原理。

答：两种for循环，通过标记变量i的递增访问每个元素，实现遍历。我之前在项目组的时候用的最多的就是遍历的思想，写封装，cms，增删改查之类的 都涉及遍历

4.如何实现响应式布局，原理是什么，做响应式布局的时候遇到哪些问题？

5.怎样添加、移除、替换、复制、创建和查找节点

答：添加appendChild()，移除removeChild()，替换replaceChild()，复制cloneNode(true/false)，创建createElement()/createTextNode()，查找getElementById()/getElementsByTagName()/getElementByName()

五组

1. new 操作符具体干了什么呢?

创建了一个空对象、连接到原型函数、绑定this值、返回新的对象

1. javascript继承的 6种方法

寄生组合式继承、还有构造函数继承、多重继承，原型链继承，我基本用到的就这几种

三、ajax 过程

六组

1. 闭包的优缺点

可以读取函数内部变量、让变量始终保持在内存中。缺点就是容易造成内存泄漏

1. 什么情况会造成缓存泄露，如何避免。
2. 你如何对网站的文件及资源进行优化？

文件合并吧，注意代码规范，比如css引入写在最上边，js写最下边，都用外部引入方式

1. Window.onload 和 document.ready 的区别。

Window.onload是网页全都加载完执行，不能写多个否则后边会覆盖前面

Document.ready是dom结构绘制完就开始执行，可以同时加载多个

Websocket和Ajax的区别