7.事件委托是什么

让利用事件冒泡的原理，让自己的所触发的事件，让他的父元素代替执行

8.闭包是什么，有什么特性，对页面有什么影响?简要介绍你理解的闭包

闭包就是能够读取其他函数内部变量的函数

2. 你有哪些性能优化的方法？

减少http请求次数，压缩css,js,图片，减少dom操作次数。优化页面结构，避免内联样式和使用css表达式，使用表格等

1.浏览器的内核分别是什么?

IE: trident内核

Firefox：gecko内核

Safari：webkit内核

Opera：以前是presto内核，Opera现已改用Google Chrome的Blink内核

Chrome：Blink(基于webkit，Google与Opera Software共同开发)

web前端开发，如何提高页面性能优化？

内容方面：减少http请求，减少dom数量

Css方面：独立css并在head内引入，精简压缩，避免css表达式

Js方面：独立js并在页面底部加载，压缩

图片方面：恰当使用图片格式，压缩。背景小图合并

1. Node的特点（优点和缺点）

Node的特点是异步非IO阻塞、适合高并发，但不适合密集型计算

ECMAScript 标准定义了 6 种数据类型

Number, String, Boolean, Null, Undefined, Object

ES6常用新特性

1. 变量声明const和let
2. 模板字符串
3. 箭头函数
4. Promise

常见算法（数组去重，统计字符出现次数，排序，随机生成指字长度字符）

1. 数组去重

**var *array*** = [1, 2, 3, 4, 4, 5, 6, 8, 2, 3, 4];  
**var *temp*** = [];  
**for** (**var *i* in *array***) {  
 **if**(***temp***.indexOf(***array[i]***)==-1){  
 ***temp***.push(***array[i]***);  
 }  
}  
**console**.log(***temp***);

1. 统计出现最多的字符

**var *string*** = **"afjghdfraaaasdenas"**;  
**var *temp*** = {};  
**var *max*** = 0;  
**for** (**var *i* in *string***) {  
 **if** (***temp***[***string***[***i***]] == **undefined**) {  
 ***temp***[***string***[***i***]] = 1  
 } **else** {  
 ***temp***[***string***[***i***]]++  
 }  
}  
**for** (**var *i* in *temp***) {  
 **if** (***temp***[***i***] > ***max***) {  
 ***max*** = ***temp***[***i***]  
 }  
}  
**console**.log(***max***);

1. 随机生成字符串

toString(36)//0-9,a-Z的字符串

Math.random().toString(36).substr(0,2)//11位

1. 排序(冒泡排序)
2. **var *arr*** = [3, 44, 38, 5, 47, 15, 36, 26, 27, 2, 46, 4, 19, 50, 48];  
   **var *len*** = ***arr***.**length**;  
   **for** (**var *i*** = 0; ***i*** < ***len***; ***i***++) {  
    **for** (**var *j*** = 0; ***j*** < ***len*** - 1 - ***i***; ***j***++) {  
    **if** (***arr***[***j***] > ***arr***[***j*** + 1]) {*//相邻元素两两对比* **var *temp*** = ***arr***[***j*** + 1];*//元素交换* ***arr***[***j*** + 1] = ***arr***[***j***];  
    ***arr***[***j***] = ***temp*** }  
    }  
   }  
   **console**.log(***arr***);

原型是什么

在Javascript中原型是一个prototype对象，用于表示类型之间的关系

原型链是什么

对象之间的继承关系，在Javascript中是通过prototype对象指向父类对象，直到指向Object对象为止，这样就形成一个原型指向的链条，称之为原型链