1. 原型链

在编写JavaScript代码时，不要直接用obj.\_\_proto\_\_去改变一个对象的原型，并且，低版本的IE也无法使用\_\_proto\_\_。Object.create()方法可以传入一个原型对象，并创建一个基于该原型的新对象，但是新对象什么属性都没有，因此，我们可以编写一个函数来创建xiaoming：  
 \_\_proto\_\_是**每个对象**都有的一个属性，而prototype是**函数**才会有的属性

// 原型对象:

var Student = {

name: 'Robot',

height: 1.2,

run: function () {

console.log(this.name + ' is running...');

}

};

function createStudent(name) {

// 基于Student原型创建一个新对象:

var s = Object.create(Student);

// 初始化新对象:

s.name = name;

return s;

}

var xiaoming = createStudent('小明');

xiaoming.run(); // 小明 is running...

或者  
 var xiaoming = Object.create(Student)

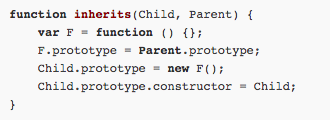
xiaoming.name = '小明'

xiaoming.run()

第二种:proto

var xiaoming = {}  
 xiaoming.\_\_prroto\_\_ = Student  
 xiaoming.name = ‘小明’  
 xiaoming.run()

第三种:空函数





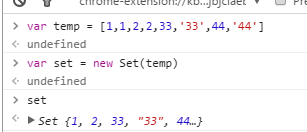
1.2  
b()返回undefined  
b.call(a)返回7

2.1数组去重

filter:

var r,arr = ['apple', 'strawberry', 'banana', 'pear', 'apple', 'orange', 'orange', 'strawberry'];  
r=arr.filter(function(element,index,self){  
    return self.indexOf(element) == index;     //indexOf只返回元素在数组中第一次出现的位置，如果与元素位置不一致，说明该元素在前面已经出现过，是重复元素,self表示原来数组自己  
})  
alert(r.toString());

set:



2.2字符串转换

times.map( (item,index)=>{ //月份 日 补成双数

let f2 = ''

let f3 = ''

if( item.split('-')[1] < 10 ){

f2 = '0'+item.split('-')[1]

}else{

f2 = item.split('-')[1]

}

if( item.split('-')[2] < 10 ){

f3 = '0'+item.split('-')[2]

}else{

f3 = item.split('-')[2]

}

return item.split('-')[0] +'-'+ f2+'-'+f3

})

3.第一种:margin 0 auto,或绝对定位

第二种: .box2{

    display: flex;

    justify-content:center;

    align-items:Center;

}

6. 在页面创建dom节点的方法

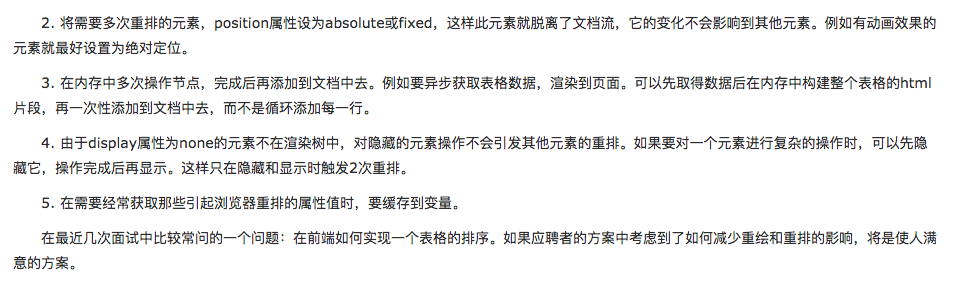
使用诸如createElement()和appendChild()之类的DOM方法，以及使用innerHTML。对于小的DOM更改，两者效率差不多，但对于大的DOM更改，innerHTML要比标准的DOM方法创建同样的DOM结构快得多。

当使用innerHTML设置为某个值时，后台会创建一个HTML解释器，然后使用内部的DOM调用来创建DOM结构，而非基于JS的DOM调用。由于内部方法是编译好的而非解释执行，故执行的更快。

7.减小重排和缩小重排影响的方法

<https://www.cnblogs.com/gyjWEB/p/4547140.html>





8. 插入100个div 每10个加个空格

