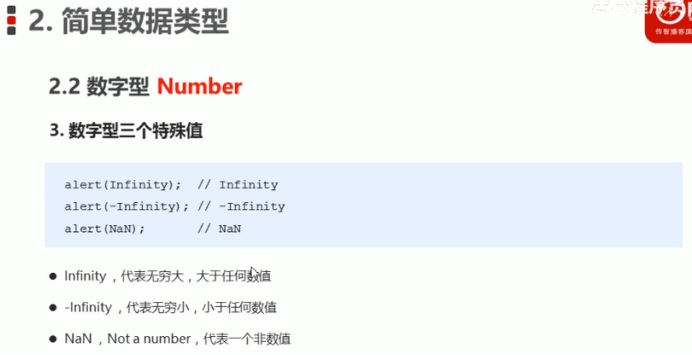
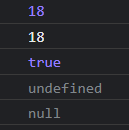
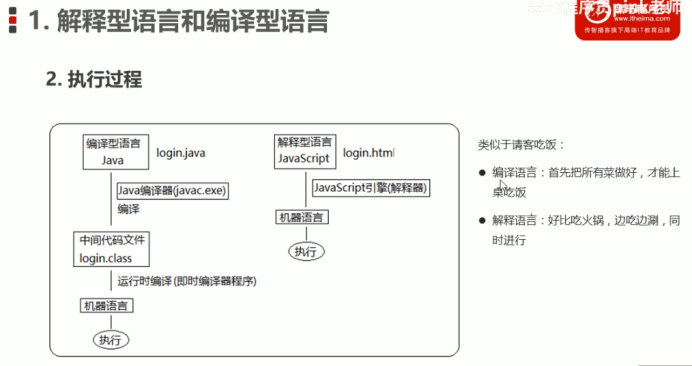
1. 
2. 
3. 
4. console.log('12' + 12);
5. var variable = undefined; / var variable = null;

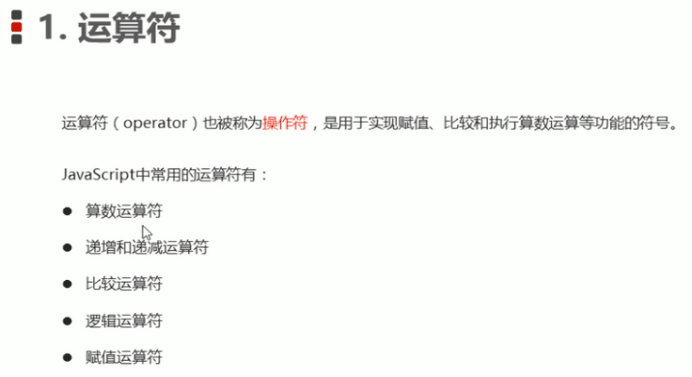
console.log(variable + 1);

console.log(variable + 'pink');

1. 
2. 转为字符串类型的三种方式
3. ‘12’- 0 得到的是啥
4. console.log('123'- '120');
5. 怎样转换成布尔型
6. 布尔值为假的几种情况
7. 什么是解释语言，什么是编译型语言，他们的区别



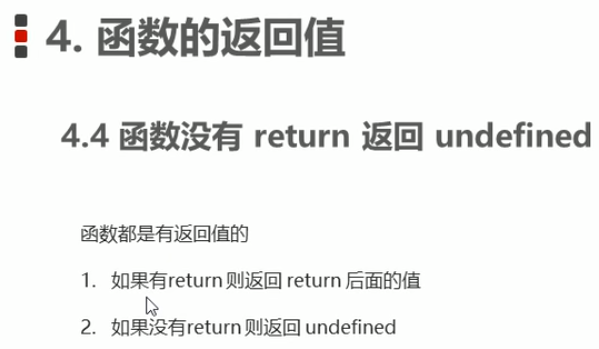
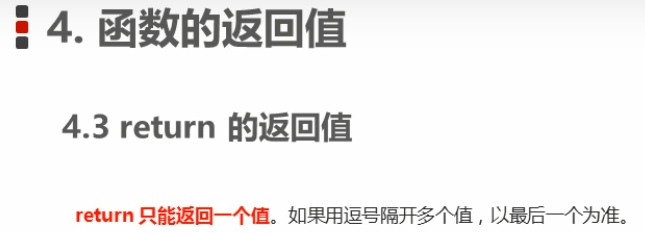
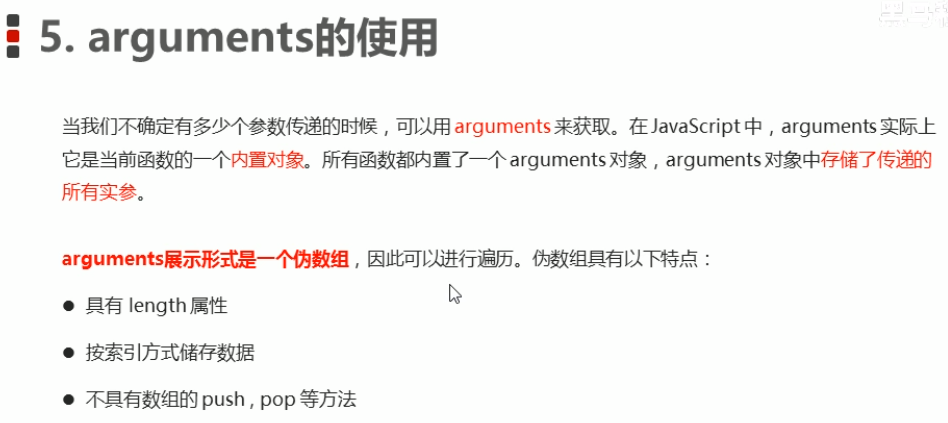
1. 关键字、标识符、保留字都是什么，区别
2. 运算符有哪些，优先级





1. 怎么解决浮点数精度问题
2. 如何判断 0.6 + 0.1 等不等于 0.7
3. ++a 和 a++ 的区别
4. var e = 10;

var f = e++ + ++e;

1. 说明短路运算的原理
2. console.log(123 && 456);
3. var num = 0;  
    console.log(123 || num++);  
    console.log(num);
4. 流程控制的三种结构
5. switch 和 if else if 区别
6. 如何断点调试
7. 
8. 
9. 
10. 
11. 
12. console.log(Math.abs('-1'));
13. 我们想要得到两个数之间的随机整数 并且 包含这2个整数

Math.floor(Math.random() \* (max - min + 1)) + min;

1. var date1 = +new Date(); // +new Date() 返回的就是总的毫秒数
2. 检测是否为数组

console.log(arr instanceof Array);

console.log(Array.isArray(arr));

1. arr1.sort(function(a, b) {

// return a - b; // 升序的顺序排列

return b - a; // 降序的顺序排列

});

1. 
2. console.log(arr1.join('-')); // green-blue-pink
3. 
4. 排他

// 1. 获取所有按钮元素

        var btns = document.getElementsByTagName('button');

        // btns得到的是伪数组  里面的每一个元素 btns[i]

        for (var i = 0; i < btns.length; i++) {

            btns[i].onclick = function() {

                // (1) 我们先把所有的按钮背景颜色去掉

                for (var j = 0; j < btns.length; j++) {

                    btns[j].style.backgroundColor = '';

                }

                // (2) 然后才让当前的元素背景颜色为pink

                this.style.backgroundColor = 'pink';

            }

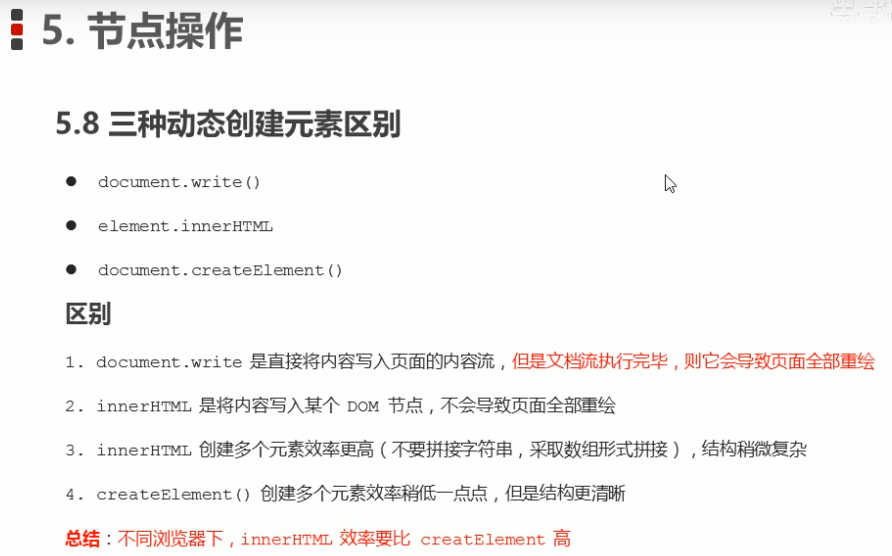
        }

        // 2. 首先先排除其他人，然后才设置自己的样式 这种排除其他人的思想称为排他思想

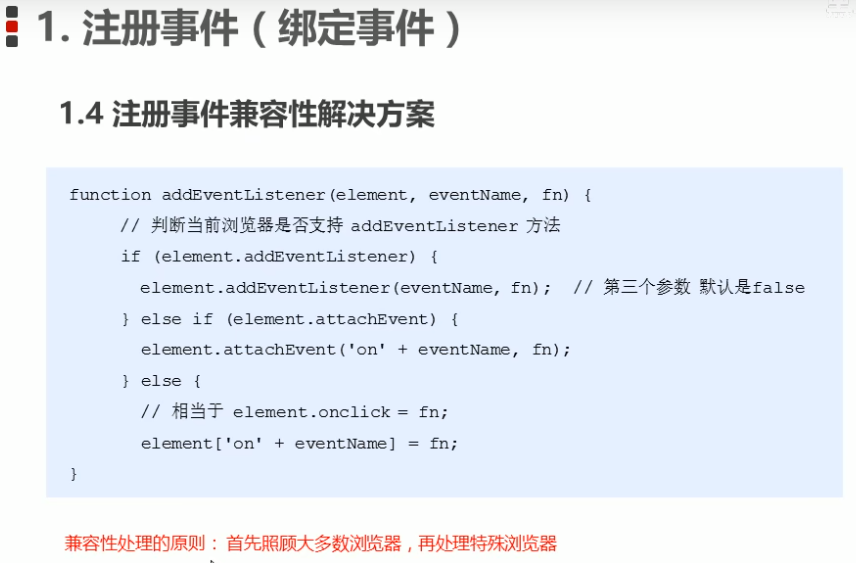
1. dataset 是一个集合里面存放了所有以data开头的自定义属性

如果自定义属性里面有多个-链接的单词，我们获取的时候采取 驼峰命名法

1. 
2. 三种动态创建元素的区别



1. 注册事件的两种方法
2. 事件监听 attchEvent



1. 事件监听删除

divs[0].onclick = function() {

            alert(11);

            // 1. 传统方式删除事件

            divs[0].onclick = null;

        }

        // 2. removeEventListener 删除事件

        divs[1].addEventListener('click', fn)  // 里面的fn 不需要调用加小括号

        function fn() {

            alert(22);

            divs[1].removeEventListener('click', fn);

        }

        // 3.

        divs[2].attachEvent('onclick', fn1);

        function fn1() {

            alert(33);

            divs[2].detachEvent('onclick', fn1);

        }

1. 捕获阶段 如果addEventListener 第三个参数是 true 那么则处于捕获阶段 body -> father -> son；冒泡阶段 如果addEventListener 第三个参数是 false 或者 省略 那么则处于冒泡阶段 son -> father -> html -> document
2. 事件对象



1. 事件委托e.target.style.backgroundColor = 'pink';
2. X和Y坐标

// 1. client 鼠标在可视区的x和y坐标

            console.log(e.clientX);

            console.log(e.clientY);

            console.log('--------------------------');

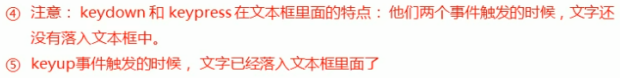
            // 2. page 鼠标在页面文档的x和y坐标

            console.log(e.pageX);

            console.log(e.pageY);

            console.log('--------------------------');

            // 3. screen 鼠标在电脑屏幕的x和y坐标

1. console.log(e.screenX);
2. console.log(e.screenY);
3. onkeydown 和 onkeypress 的区别，执行顺序以及大小写区分
4. 
5. 三个事件的执行顺序 keydown -- keypress -- keyup
6. 我们的keyup 和keydown事件不区分字母大小写 a 和 A 得到的都是65  
   我们的keypress 事件 区分字母大小写
7. 
8. load 等页面内容全部加载完毕，包含页面dom元素 图片 flash css 等等  
     DOMContentLoaded 是DOM 加载完毕，不包含图片 flash css 等就可以执行
9. this 指向问题
10. 全局作用域或者普通函数中this指向全局对象window(注意定时器里面的this指向window)
11. 方法调用中谁调用this指向谁
12. 构造函数中this指向构造函数的实例
13. JS执行机制
14. location方法实现页面跳转

            // 记录浏览历史，所以可以实现后退功能

            // location.assign('http://www.itcast.cn');

            // 不记录浏览记录，所以不可以实现后退功能

            // location.replace('http://www.itcast.cn');

1. offset + client + scroll 三大系列

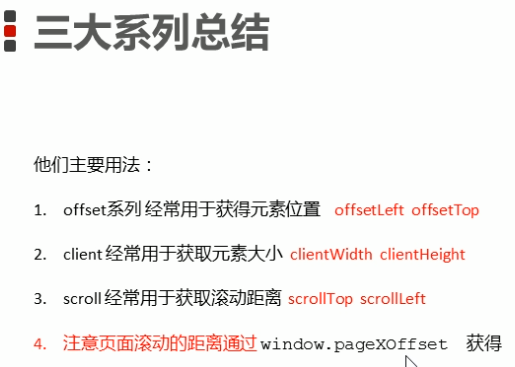




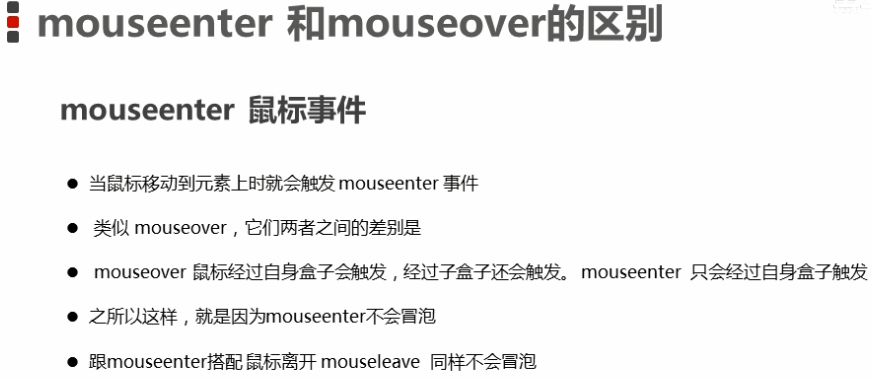








1. 立即执行函数 (function() {})() 或者 (function(){}());
2. mouseenter 和 mouseover 的区别



1. 触摸事件



1. 类名增删改查

  // classList 返回元素的类名

        var div = document.querySelector('div');

        // console.log(div.classList[0]);

        // 1. 添加类名  是在后面追加类名不会覆盖以前的类名 注意前面不需要加.

        div.classList.add('three');

        // 2. 删除类名

        div.classList.remove('one');

        // 3. 切换类

        var btn = document.querySelector('button');

        btn.addEventListener('click', function() {

            document.body.classList.toggle('bg');

        })

1. 本地存储两种方式





1. 边框图片



1. Echarts