传智播客·前端与移动开发学院
http://web.itcast.cn

前端基本功—javascript 第十一天

目录

1.1 复习

1. clientX　　clientWidth　可视区域的宽度

   clientWidth　　width　+　padding

   offsetWidth　　width + padding + border

   scrollWidth　　width + padding　　超过　内容的宽度

2. window.onresize　　窗口改变

3. 匀速运动函数　　原理 ：　盒子的原来位置　+　步长

1.2 三个取整函数

这三个函数都是　数学函数

Math

Math.ceil()　　向上取整　　天花板

比如说　console.log(Math.ceil(1.01))　　结果 是 2

　　console.log(Math.ceil(1.9))　　结果 2

　　console.log(Math.ceil(-1.3))　　结果 是　-1

Math.floor()　向下取整　　地板

　比如说 console.log(Math.floor(1.01))　　结果 是 1

　　console.log(Math.floor(1.9))　　结果 1

　　console.log(Math.floor(-1.3))　　结果 是　-2

Math.round()　四舍五入函数

　　console.log(Math.round(1.01))　　结果 是 1

　　console.log(Math.round(1.9))　　结果 是 2

1.3 缓动动画原理

匀速动画的原理：　盒子本身的位置　+　步长

缓动动画的原理:　盒子本身的位置　+　步长 (不断变化的)

封装代码：

```
function animate(obj,target){    //    第一个参数  动谁      第二个参数    动多少
        clearInterval(obj.timer);
        obj.timer = setInterval(function() {
                // 计算步长      动画的原理        盒子本身的位置    +    步长
                var step = (target - obj.offsetLeft) / 10;    // 步长
                step =    step > 0 ? Math.ceil(step) : Math.floor(step);    //    取整步长
                // obj.style.left =  盒子本身的位置    +    步长
                obj.style.left = obj.offsetLeft + step + "px";
                if(obj.offsetLeft == target){
                        clearInterval(obj.timer);
                  }
          },30)
    }
```

1.4　js 常用 访问 CSS 属性

我们访问得到 css 属性，比较常用的有两种：

1. 利用点语法

box.style.width box.style.top

点语法可以得到 width 属性 和 top 属性 带有单位的。 100px

但是这个语法有非常大的缺陷， 不变的。

后面的 width 和 top 没有办法传递参数的。

var w = width;

~~box.style.w~~

2. 利用 [] 访问属性

语法格式: box.style[“width”]

元素.style[“属性”];

```
console.log(box.style["left"]);
```

最大的优点 ： 可以给属性传递参数

1.5 得到 css 样式

我们想要获得 css 的样式， box.style.left box.style.backgorundColor

但是它只能得到 行内的样式。

但是我们工作最多用的是 内嵌式 或者 外链式 。

怎么办？

核心： 我们怎么才能得到内嵌或者外链的样式呢？

1. obj.currentStyle ie opera 常用

外部（使用<link>）和内嵌（使用<style>）样式表中的样式（ie 和 opera）

2 .window.getComputedStyle("元素", "伪类") w3c

两个选项是必须的， 没有伪类 用 null 替代

3 兼容写法 ：

我们这个元素里面的属性很多， left top width ===

我们想要某个属性， 就应该 返回改属性，所有继续封装 返回当前样式的 函数。

```
var demo = document.getElementById("demo");
    function getStyle(obj,attr) {   //   谁的        那个属性
        if(obj.currentStyle)   // ie 等
        {
            return obj.currentStyle[attr];
        }
        else
        {
            return window.getComputedStyle(obj,null)[attr];   // w3c 浏览器
        }
    }
    console.log(getStyle(demo,"width"));
```

1.6 JSON 遍历

for in 关键字

for (变量 in 对象)

{ 执行语句; }

```
var json = {width:200,height:300,left:50}
console.log(json.width);
for(var k in json)
{
    console.log(k);    // k 遍历的是 json  可以得到的是属性
    console.log(json[k]);  // json[k]  得到 是属性的值
}
```

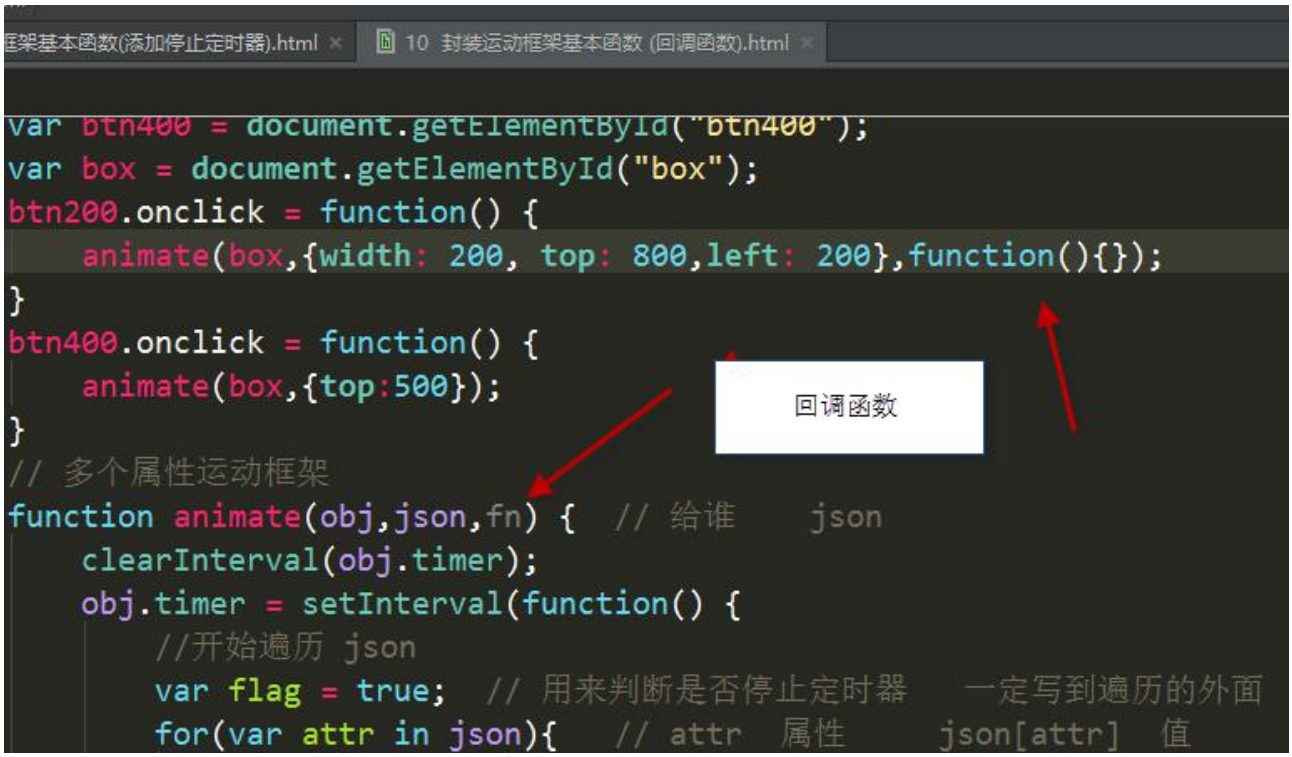
千万要记得 每个 的意思 ： 那是相当重要

k 是 属性

json[k] 得到的是属性值

1.6.1 回调函数

等动画执行完毕再去执行的函数　　回调函数



有个简单的问题？ 什么是爱情？

我们怎么知道动画就执行完毕了呢？

很简单　 当定时器停止了。 动画就结束了

1.7 in 运算符

in 运算符也是一个二元运算符，但是对运算符左右两个操作数的要求比较严格。in 运算符要求第 1 个（左边的）操作数必须是字符串类型或可以转换为字符串类型的其他类型，而第 2 个（右边的）操作数必须是数组或对象。只有第 1 个操作数的值是第 2 个操作数的属性名，才会返回 true，否则返回 false

```
// in 可以用用来判断 json 里面有没有某个属性
```

```
var json = {name: "刘德华",age : 55};
// in 可以用用来判断 json 里面有没有某个属性
if("andy" in json)
{
    console.log("yes");  // 返回的是 yes
}
else
{
    console.log("no");
}
```