



传智播客 · 前端与移动开发学院
<http://web.itcast.cn>

前端基本功—javascript 第十天

目录

目录.....	2
1.2 复习.....	3
1.3 client 家族.....	3
1.3.1 检测屏幕宽度(可视区域).....	4
1.3.2 window.onresize 改变窗口事件.....	5
1.4 检测屏幕宽度(分辨率).....	5
1.5 简单冒泡机制.....	6
1.5.1 阻止冒泡的方法.....	7
1.5.2 小案例 点击空白处隐藏盒子.....	7
1.5.3 判断当前对象.....	8
1.5.4 选中之后，弹出层.....	11
1.5.5 获得用户选择内容.....	11

1.1 复习

1. window - document
2. document.title document.head document.body
3. document.documentElement (约等于 ~~document.html~~)
4. scrollTop 被卷去的头部 scrollLeft 封装自己的 函数
5. scrollTo(x,y) 去往页面的 x 和 y 坐标 的位置
6. window.scrollTo(x,y)
7. window.onscroll = fucntion() { fun (); } fun() function()
{iffss }
8. JSON js 对象表示法 数据传输
9. var json = {} 对象 var arr = []; 数组 var num; 变量
10. var json = { width: "100px", height: 100 } 声明的方法
11. json.height 100 使用

1.2 client 家族

client 可视区域

offsetWidth: width + padding + border (披着羊皮的狼)

clientWidth: width + padding 不包含 border

scrollWidth: width + padding 大小是内容的大小 (如果超

过盒子大小, 没超过则是 width + padding)



1.2.1 检测屏幕宽度(可视区域)

ie9 及其以上的版本

`window.innerWidth`

标准模式

`document.documentElement.clientWidth`

怪异模式

`document.body.clientWidth`

自己封装一个 返回可视区宽度和高度的函数。

1.2.2 window.onresize 改变窗口事件

昨天 `window.onscroll = function() {}` 屏幕滚动事件

今天 `window.onresize = function() {}` 窗口改变事件

`onresize` 事件会在窗口或框架被调整大小时发生

要求:

当我们的页面宽度大于 960 像素的时候 页面颜色是红色

当我们的页面宽度 大于 640 小于 960 页面的颜色是 绿色

剩下的颜色是 蓝色

`function fun() { 语句 }`

`fun` 是函数体的意思

`fun()` 调用函数 的意思

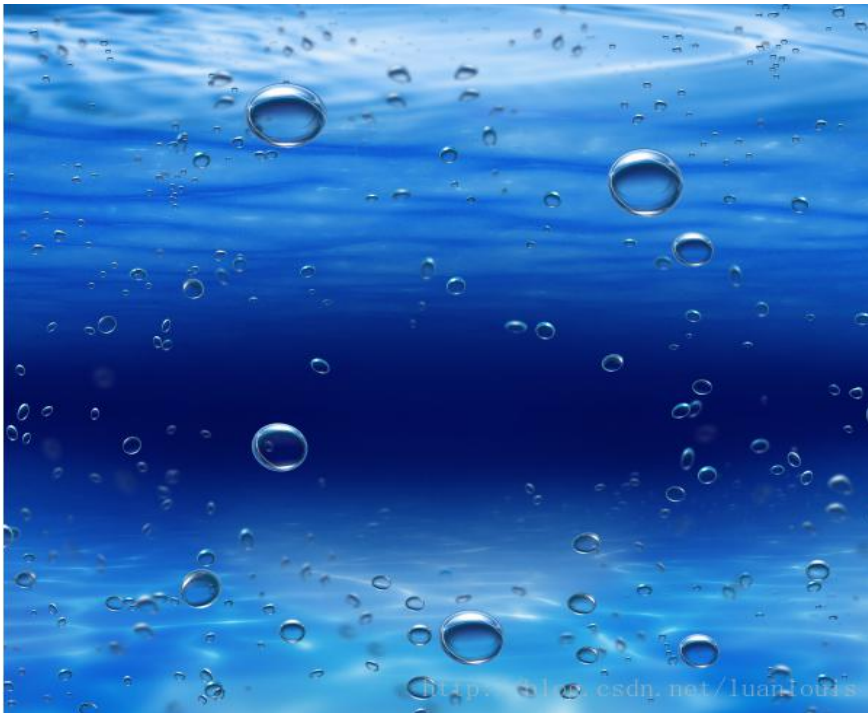
```
function fun() {  
    return 3;  
}  
console.log(fun); // 返回函数体 function fun() { retrun 3}  
console.log(fun()); // 调用函数 3 返回的是结果  
fun();  
window.onresize = 3  
window.onresize = function fun() { retrun 3}
```

1.3 检测屏幕宽度(分辨率)

`clientWidth` 返回的是 可视区 大小 浏览器内部的大小

`window.screen.width` 返回的是我们电脑的 分辨率 跟浏览器没有关系

1.4 简单冒泡机制



事件冒泡: 当一个元素上的事件被触发的时候, 比如说鼠标点击了一个按钮, 同样的事件将会在那个元素的所有祖先元素中被触发。这一过程被称为事件冒泡; 这个事件从原始元素开始一直冒泡到 DOM 树的最上层。

顺序

E 6.0:

`div -> body -> html -> document`

其他浏览器:

`div -> body -> html -> document -> window`

不是所有的事件都能冒泡。以下事件不冒泡: `blur`、`focus`、`load`、`unload`

1.4.1 阻止冒泡的方法

标准浏览器 和 ie 浏览器

w3c 的方法是 `event.stopPropagation()`

`propagation` 传播 传递

IE 则是使用 `event.cancelBubble = true`

`bubble` 冒泡 泡泡

`cancel` 取消

兼容的写法:

```
1  if(event && event.stopPropagation)
2      {
3          event.stopPropagation(); // w3c 标准
4      }
5      else
6      {
7          event.cancelBubble = true; // ie 678 ie 浏览器
8      }
```

1.4.2 小案例 点击空白处隐藏盒子

这个案例就是说，一个盒子，点击除了自己之外的任何一个地方，就会隐藏。

原理:

点击自己不算 (怎么证明我是我 点击的这个对象 `id` 正好和自己一样)

点击空白处 就是点击 `document`

1.4.3 判断当前对象

火狐 谷歌 等 `event.target.id`

ie 678 `event.srcElement.id`

兼容性写法:

```
var targetId = event.target ? event.target.id : event.srcElement.id;
```

```
targetId != "show"
```

代码:

```
1 <!DOCTYPE html>
2 <html>
3 <head lang="en">
4     <meta charset="UTF-8">
5     <title></title>
6     <style>
7         body {
8             height:2000px;
9         }
10        #mask {
11            width: 100%;
12            height: 100%;
13            opacity: 0.4;    /*半透明*/
14            filter: alpha(opacity = 40);    /*ie 6 半透明*/
```



```
15         background-color: black;
16         position: fixed;
17         top: 0;
18         left: 0;
19         display: none;
20     }
21     #show {
22         width: 300px;
23         height: 300px;
24         background-color: #fff;
25         position: fixed;
26         left: 50%;
27         top: 50%;
28         margin: -150px 0 0 -150px;
29         display: none;
30     }
31 </style>
32 </head>
33 <body>
34 <a href="javascript:;" id="login">注册</a>
35 <a href="javascript:;">登录</a>
36 <div id="mask"></div>
37 <div id="show"></div>
38 </body>
39 </html>
40 <script>
41     function $(id) { return document.getElementById(id);}
42     var login = document.getElementById("login");
43     login.onclick = function(event) {
```

```
44      $(".mask").style.display = "block";
45      $(".show").style.display = "block";
46      document.body.style.overflow = "hidden"; // 不显示滚动条
47
48      //取消冒泡
49      var event = event || window.event;
50      if(event && event.stopPropagation)
51      {
52          event.stopPropagation();
53      }
54      else
55      {
56          event.cancelBubble = true;
57      }
58      document.onclick = function(event) {
59
60          var event = event || window.event;
61
62          // alert(event.target.id); // 返回的是点击的某个对象的id 名字
63          // alert(event.srcElement.id);
64          var targetId = event.target ? event.target.id : event.srcElement.id;
65
66          // 看明白这个写法
67          if(targetId != "show") // 不等于当前点点击的名字
68          {
69              $(".mask").style.display = "none";
70              $(".show").style.display = "none";
71
72              document.body.style.overflow = "visible"; // 显示滚动条
```

```
70      }  
71    }  
72  </script>
```

1.4.4 选中之后，弹出层

我们想，选中某些文字之后，会弹出一个弹出框

这个和 我们前面讲过的拖拽有点不一样。

拖拽 是拖着走。 拉着鼠标走 。

选择文字： 这个弹出的层 选中的时候不出来，弹起鼠标的时候才出现 。

所以这个的事件一定是 `onmouseup` 。 盒子显示而且盒子的位置 再 鼠标的 `clientX` 和 `clientY` 一模一样

用来判断选择的文字：

1.4.5 获得用户选择内容

`window.getSelection()` 标准浏览器

`document.selection.createRange().text;` ie 获得选择的文字

兼容性的写法：

```
if(window.getSelection)  
{  
    txt = window.getSelection().toString(); // 转换为字符串  
}  
else  
{  
    txt = document.selection.createRange().text; // ie 的写法  
}
```

综合代码：

```

1  <!DOCTYPE html>
2  <html>
3  <head lang="en">
4      <meta charset="UTF-8">
5      <title></title>
6      <style>
7          div {
8              width: 400px;
9              margin:50px;
10         }
11         #demo {
12             width: 100px;
13             height: 100px;
14             background-color: pink;
15             position: absolute;
16             top: 0;
17             left: 0;
18             display: none;
19         }
20     </style>
21 </head>
22 <body>
23     <span id="demo"></span>
24     <div id="test">我要复制的文字我要复制的文字我要复制的文字我要复制的文字我要复

```

```
25  <div id="another">
26      我不要的文字 我不要的文字 我不要的文字 我不要的文字 我不要的文字 我不要
    的文字 我不要的文字 我不要的文字 我不要的文字 我不要的文字 我不要的文字 我不要的文
    字 我不要的文字 我不要的文字 我不要的文字 我不要的文字 我不要的文字 我不要的文字
27  </div>
28  </body>
29  </html>
30  <script>
31      function $(id) {return document.getElementById(id)}
32      $("#test").onmouseup = function(event) {
33          var event = event || window.event;
34          var x = event.clientX; // 鼠标的x 坐标
35          var y = event.clientY; // 同理
36          var txt; // 用于存贮文字的变量
37          if(window.getSelection) // 获取我们选中的文字
38              {
39                  txt = window.getSelection().toString(); // 转换为字符串
40              }
41          else
42              {
43                  txt = document.selection.createRange().text; // ie 的写法
44              }
45          if(txt) // 所有的字符串都为真 "" 是假 所有的数字为真 0 为假
46              {
```

```
47          //看看有没有选中的文字，没有选中文字为空的，就不应该执行  点击一
下鼠标 就是空的

48          showBox(x,y,txt);  // 调用函数
49      }
50  }

51  document.onclick = function(event) {  // 点击空白处隐藏

52      var event = event || window.event;
53      var targetId = event.target ? event.target.id : event.srcElement.id;
54      if(targetId != "demo"){
55          $("demo").style.display = "none";
56      }
57  }

58  function showBox(mousex,mousey,contentText) {  // 相关操作

59      setTimeout(function(){
60          $("demo").style.display = "block";
61          $("demo").style.left = mouseX + "px";
62          $("demo").style.top = mousey + "px";
63          $("demo").innerHTML = contentText;
64      },300);
65  }
66  </script>
```

1.5 动画原理

人走路的时候， 步长

动画的基本原理：让盒子的 `offsetLeft` + 步长

盒子 原来的位置 0 + 10 盒子现在的 `offsetLeft` 10

$10 + 10 = 20 + 10$

原理：

```
<script>
//动画的基本原理 盒子的 offsetLeft + 步长
var btn = document.getElementsByTagName("button")[0];
var div = document.getElementsByTagName("div")[0];
var timer = null;
btn.onclick = function() {
    timer = setInterval(function() {
        if(div.offsetLeft > 400)
        {
            clearInterval(timer);
        }
        div.style.left = div.offsetLeft + 10 + "px";
    },20);
}
</script>
```

$|-5| = 5$

`Math.abs(-5)` 取绝对值函数 js 就是 数学计算

1.5.1 匀速运动封装函数

```
1 function animate(obj,target){
2     var speed = obj.offsetLeft < target ? 5 : -5; // 用来判断 应该 + 还是 -
3     obj.timer = setInterval(function() {
4         var result = target - obj.offsetLeft; // 因为他们的差值不会超过5
5         obj.style.left = obj.offsetLeft + speed + "px";
6         if(Math.abs(result)<=5) // 如果差值不小于 5 说明到位置了
7         {
```

```
8             clearInterval(obj.timer);  
9             obj.style.left = target + "px"; // 有5 像素差距 我们直接跳转目标位置  
10          }  
11      },30)  
12  }
```