传智播客·前端与移动开发学院
http://web.itcast.cn

前端基本功—javascript 第五天

目录

1.1 复习

1. 循环

for(初始化; 退出条件; 增量)　{　}

while(退出条件) {　　}

do {　语句 }　while(退出条件)

2. switch( )　多选 1

```
switch(参数)
{
    case “参数 1”：
    语句;
    break;
  case “参数 2”：
    语句;
    break;
default:

}
```

3. 数组常用的方法

加内容　减内容　　连接　　转换

push();　　向数组的后面添加内容

var arr = [1,2]　　arr.push(“你好吗”);　　结果 [1,2,”你好吗”];

unshift()　数组的前面添加

var arr = [1,2]　　arr.unshift(“我很好”)　结果　[“我很好”,1,2]’

删除

1. pop()　　删除最后一个元素

var arr = [1,2]　　arr.pop();　　[1]

2. shift()　删除第一个元素

var　arr = [1,2]　　arr.shift()　　[2]

注意：

var arr = [1,2,3,4];

console.log(arr.push(8));　　结果是　5　　返回改数组的长度

如果这么写：

var arr = [1,2,3,4];

arr.push(8);

console.log(arr);　　结果 就是　【1,2,3,4,8】

var arr = [1,2,3,4,5,6,12];

console.log(arr.pop());　　结果是 12　返回最后一个元素

var arr = [1,2,3,4];

arr.pop();

console.log(arr);　　结果 就是　【1,2,3】

连接　　concat

join()　　把数组转换为字符串

var　arr = [1,2,3,4];

var test = arr.join(“-”);

console.log(test)　　1-2-3-4

var arr = [1,2,3,4];

arr.join(“*”);

console.log(arr);　　[1,2,3,4]

split()　把字符串转换为 数组

var　txt = “1-2-3”;

var　test = txt.split(“-”)

console.log(test)　　[“1”,”2”,”3”];

4. dom

我们操作最多的就是　元素节点　　标签节点　　标签　　li span

5. 节点的访问关系

父级　　parentNode

this.parentNode　　==　我的父亲

兄弟　　nextSibling

孩子们

childNodes　　官方用法

一般情况下，我们只需要元素节点

nodeType　来　判断

nodeType == 1 元素节点

nodeType == 2 属性节点

nodeType == 3 文本节点

 内容

children 不是官方写法 所有的孩子 亲儿子

ie 678 把注释节点 也算 可以避免

1.2 设置节点属性

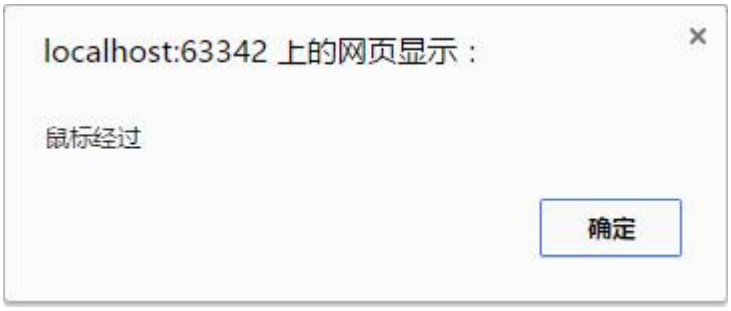
1. 获取节点属性

getAttribute(属性) 获取属性

通过这个方法，可以得到 某些元素的 某些属性 。

```
alert(demo.getAttribute("title"));
```

弹出对话框： 弹出 title 里面的内容



2. 设置节点属性

setAttribute(“属性”,”值”);

比如说，我们想要把 一个 类名 改为 demo

div.setAttribute(“class”,”demo”);

3. 删除某个属性

removeAttribute(“属性”);

demo.removeAttribute(“title”)

这个盒子就没有 title　属性　给删掉了　。

A.appendChild(B);

B 一定是 A 孩子　　同时　b 放到了 a 的里面　装到里面去了最后面。　b 放到　a 里面

A.insertBefore（B，C）

B　C　都是 A 的孩子

把 b　放到　a 里面 ，但是 是 c 的前面

1.3　内置对象

内置对象：　内置对象就是指这个语言自带的一些对象，供开发者使用，这些对象提供了一些常用的或是最基本而必要的功能。

手机买来就能发短信　　就能打电话

1.4 日期函数 （Date()）

这个函数 （对象） 可以设置我们本地 日期。 年月日 时分秒

1.4.1 声明日期

```
var date = new Date();
```

创造声明一个新的日期函数 赋值给了 date

var arr = new Array();

1.4.2 使用函数

得到 毫秒数 ms s m h

从 1970 年 1 月 1 号 unix 32 位 68 年

2038 千年虫 64 位

var date = new Date();

date.getTime();

date.valueOf(); 得到 距离 1970 年的毫秒数

```
var date = new Date(); // 声明

console.log(date.getTime()); // 提倡使用的
console.log(date.valueOf());

// 直接使用
console.log(Date.now());
console.log(+new Date());
```

1.5 常用的日期的方法

获取日期和时间

getDate()　　获取日 1-31

getDay ()　　获取星期 0-6

getMonth ()　　获取月　0-11

getFullYear ()　　获取完整年份（浏览器都支持）

getHours ()　　获取小时 0-23

getMinutes ()　　获取分钟 0-59

getSeconds ()　　获取秒　0-59

getMilliseconds ()　　获取当前的毫秒

getTime ()　　返回累计毫秒数(从 1970/1/1 午夜)

如果是上午　，我打开页面　　页面中显示的是

上午好，好好学习　　显示的是上午的图片

如果是下午　我打开页面　　页面中显示的是

下午好，天天向上　　显示的是下午的图片

根据当前的小时来判断　　if

1.6 定时器

很多情况下，一些操作不需要人工干预， 代码自己会不断的去执行 。而且 会有 时间的绑定 。 比如每隔 5 秒钟就去执行一次事件。

我们可以设定时间 让某个动作不断的去执行 。 这个我们再 js 里面用定时器来表示。

```
window.setInterval(“执行的函数”,间隔时间)
```

正确的写法:

```
setInterval(fun, 1000);          1000 ms     毫秒
```

每隔 1 秒钟，就去执行一次 fun 这个函数.

setInterval(“fun()”,1000) 可以用

setInterval(function(){} , 1000)

~~setInterval(fun(),1000) 错误的~~

—

定时器 特别的像 for 循环，但是我 的定时器最大的特点在于， 自动，可以设定时间。

1.7 倒计时

好比，今年你多大了 ？

2015 - 1990 25

我们要计算的 倒计时

有一个最终时间 12 月 12 日

有一个现在时间 11 月 13 日

倒计时 = 用 将来的时间 - 现在的时间

问题： 用 毫秒减去 现在距离 1970 年 1

将来时间 距离 1970 毫秒数

用将来的毫秒数 - 现在的毫秒数 不断转换就可以了

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title></title>
    <style>
        body{
            font-size:30px;
            text-align: center;
            color:red;
        }
    </style>
    <script>
        window.onload = function(){
            var demo = document.getElementById("demo");
            var endTime = new Date("2015/12/12 17:30:00"); // 最终时间
            setInterval(clock,1000); // 开启定时器
            function clock(){
```

```
                var nowTime = new Date(); // 一定是要获取最新的时间
                //  console.log(nowTime.getTime());  获得自己的毫秒
                var second = parseInt((endTime.getTime() - nowTime.getTime()) / 1000);
                // 用将来的时间毫秒 -  现在的毫秒  / 1000 得到的 还剩下的秒  可能处不断 取整
                // console.log(second);
                 // 一小时 3600 秒
                // second / 3600  一共的小时数  /24   天数
                var d = parseInt(second / 3600 / 24);  //天数
                //console.log(d);
                var h = parseInt(second / 3600  % 24)  // 小时
                // console.log(h);
                var m = parseInt(second / 60  % 60);
                //console.log(m);
                var s = parseInt(second % 60); // 当前的秒
                console.log(s);
                /* if(d<10)
                {
                    d = "0" + d;
                }*/
                d<10 ? d="0"+d : d;
                h<10 ? h="0"+h : h;
                m<10 ? m="0"+m : m;
                s<10 ? s="0"+s : s;
                demo.innerHTML = "距离抢购时间还剩: "+d+"天 "+h+"小时 "+m+"分钟 "+s+"秒";
            }
        }
    </script>
</head>
<body>
<div id="demo"></div>
</body>
```

1.7.1 定义自己的日子

var endTime = new Date(“2015/12/12”);

如果 date 括号里面写日期 就是 自己定义的时间

如果 date 括号里面不写日期 ， 就是当前时间 。

```
new Date( “2015/12/12 17:30:00” );
```

日期和时分秒中间 有空格隔开