

JS第二周课件

 **珠峰培训**
最专业的前端培训

Node

JS

HTML5



1.1 认识DOM

DOM : document object model 文档对象模型

DOM就是描述整个html页面中节点关系的图谱

在DOM中，提供了很多获取元素的方法和之间关系的属性以及操作这些元素的方法





1.2 DOM操作

1.document.getElementById('id值')

- 注意：如果页面中id重复了，那么这个方法默认只获得第一个元素
- 在ie6、7中不区分id的大小写
- 项目实战注意：不要让表单元素的那么和其他元素的id重复，不要用id的大小写来区分不同的元素

2.document.getElementsByClassName('类名')

- 不兼容ie6~ie8，会报错

3.document.getElementsByTagName('标签名')





1.3 DOM操作

4.document.getElementsByName('name值')

- 在ie浏览器下 只对表单元素起作用，这个方法应用于具有同样name的表单元素，一般我们只用来操作表单元素

5.获取可视窗口的宽度

- document.documentElement.clientWidth||document.body.clientV

6.获取可视窗口的宽度

- document.documentElement.clientHeight||document.body.client

7.document.querySelector('.p1')

8.document.querySelectorAll('.p1')





1.4DOM操作

9.获取html元素

- document.documentElement

10.获取body元素

- document.body





2.1 认识节点

节点 (Node)

- 概念：一个页面中的标签、文字、注释、..都是节点





节点类型	nodeType	nodeName	nodeValue
------	----------	----------	-----------

2.2 节点类型

节点类型	nodeType	nodeName	nodeValue
元素节点	1	大写的标签名	null
文本节点	3	#text	文本内容
注释节点	8	#comment	注释内容
document节点	9	#document	null



3、节点关系

- 1.parentNode 父节点
- 2.childNodes 子节点
- 3.children 子元素 【不兼容】
- 4.previousSibling 上一个哥哥节点
- 5.nextSibling 下一个弟弟节点
- 6.firstChild 第一个子节点
- 7.lastChild 最后一个子节点





4、元素的动态操作

1.元素的动态创建

- 动态创建元素：document.createElement(标签名)；
- 克隆：obj.cloneNode(布尔值) true:把元素里面的子子孙孙都克隆，默认是false，只克隆当前的

2.元素的动态插入

- 父级.appendChild(要插入的元素); //把新元素插入到父容器的末尾；
- 父级.insertBefore(newEle,oldEle);把新的元素newEle添加到老的元素oldEle之前

3.元素的删除和替换

- 父级.removeChild(要删除的元素) //在父容器下删除某个元素
- 父级.replaceChild(newEle,oldEle);//在父容器下用新元素替换旧元素；



5、attribute属性

1.设置属性 `obj.setAttribute('属性名','属性值')`

2.获取属性 `obj.getAttribute('属性名')`

3.删除元素的属性，用`obj.removeAttribute('属性名');`

- IE6~8下这个方法不能修改class属性 `元素.setAttribute('class','bg')` 是不起作用的

注意事项：. 和 attribute 不能一起使用

- 通过点设置属性，就通过点来获取属性；通过`setAttribute`来设置属性，就`getAttribute`来获取属性；
- 区别：.在html中看不到 attribute可以在html中看到





6.1、数组常用方法

1.增加、删除、替换

- 1.push
- 2.pop
- 3.unshift
- 4.shift
- 5.splice splice(n,m) 删除 splice(n,0,x) 增加 splice(n,m,x) 替换
 - 使用splice 实现向数组末尾添加一项





6.2、数组常用方法

2.查找和克隆

- slice(n,m) 从索引n找到索引m，前包前不包后
 - slice(n)
 - slice(0) slice() 克隆
 - 需求：找到[第n项，第m项],原数组不发生改变： slice(n-1,m)

ary1.concat(ary2,ary3.....) 数组拼接





6.3 数组常用方法

3. 排序和排列

- sort 排序
 - sort(function(a,b){return a-b}) 从小到大
 - sort(function(a,b){return b-a}) 从大到小
- reverse() 翻转





6.4 数组常用方法

4. 数组转字符串

- toString();
- join('分隔符'); 按照分隔符转为字符串
- eval:js中 把字符串做为JS代码来执行 ;





6.5 数组常用方法

5. 常用但不兼容的方法

- indexOf() 查找：找到返回内容索引；找不到：返回-1；
- forEach(function(item,index,input){ item：数组中的每项内容
index:索引 input：原数组 });
- map:和forEach功能一样，区别就是map有返回值；





1. 删除数组最后一项：

- pop()
- splice(ary.length-1,1)
- ary.length--
- ary.length-=1;
- ary.length=ary.length-1;

2. 给数组末尾添加一项：

- push()
- splice(ary.length,0,'')
- ary[ary.length]='';

3. 克隆：slice(0)/slice(); splice(0); ary.concat();



8.Math的常用方法

1. `Math.max()` 取最大值
2. `Math.min()` 取最小值
3. `Math.ceil()` 向上取整
4. `Math.floor()` 向下取整
5. `Math.round()` 四舍五入
6. `Math.random()` 获取[0,1)之间的随机数
7. `Math.abs()` 取绝对值
8. `Math.sqrt()` 开平方
 - 获取从n-m之间的随机整数
 - 公式：`Math.round(Math.random() * (m-n) + n)`;



9.认识字符串

1.概念：在js中 在引号中的我们叫做字符串，一个字符串是由多个字符组成的

2.字符串也有 `索引`，索引是从0开始的

3.字符串也有 `length` 这个属性 获取字符串中字符的个数（字符串的长度）

空格、换行、！、，、中文等 这些特殊的符号也是 `一个字符`



10.字符串常用的方法

1.通过下标找字符

- `charAt(索引)` 通过下标找对应字符
- `charCodeAt(索引)` 通过下标找对应字符的 Unicode 编码

2.通过字符找下标

- `indexOf()`:从前往后找； 找到的时候，返回对应的索引，找不到，返回-1；
- `lastIndexOf()`：从后往前找； 找到的时候，返回对应的索引，找不到，返回-1；





10.字符串常用的方法

3.字符串的截取

- substr(n,m) 从索引n开始，截取m个；
- substring(n,m) 从索引n开始，截取到索引m，包前不包后；
- slice(n,m) slice功能跟substring一样，但是slice支持负值；

- 4.字符串的转大小写

- toLowerCase() 字符串转小写
- toUpperCase() 字符串转大写





10.字符串常用的方法

5.可以跟正则配合的字符串方法：

- 1)split('分隔符') 根据 **分隔符** 将字符串转数组
 - 如果没有对应的内容，将整体作为数组的一项返回
- 2)replace(旧,新) 字符串替换
- 3)match() 捕获：找到要找的字符，返回数组[内容，索引，原字符串]，找不到返回null
- 4)search() 查找字符：找到的时候，返回对应内容的索引，找不到返回-1；





10.1 字符串例子

1.将'2016-10-17 20:06:5'转为 2016年10月17日 20时06分05秒

2.字符串和Math的综合引用

- 实现四位不重复的验证码





11.定时器

1. `setTimeout(fn, 1000)` 1000毫秒后执行一次；而且仅仅只执行一次
2. `setInterval(fn, 500)` 每隔500毫秒执行一次
3. 定时器的返回值：具体的数字，代表是第几个定时器；
4. 定时器使用需要注意：每开启一个定时器之前，一定要先关闭之前没用的定时器；
5. 定时器关闭
 - `clearTimeout` 可以关闭`setTimeout()`；也可以关闭 `setInterval()`；
 - `clearInterval` 可以关闭`setTimeout()`；也可以关闭 `setInterval()`；



获取自己电脑的时间 `var time = new Date();`

- 标准时间格式：`Mon Nov 28 2016 20:59:36 GMT+0800 (中国标准时间)`

- `var year = time.getFullYear();` 年
- `var month = time.getMonth()+1;` 月，默认返回0~11 需要加1
- `var day = time.getDate();` 日
- `var w = time.getDay();` 星期
- `var hour = time.getHours();` 小时
- `var minutes = time.getMinutes();` 分钟
- `var seconds = time.getSeconds();` 秒
- `var mSeconds = time.getMilliseconds();` 毫秒



13、Date应用

1.将星期对应的数字，通过字符串charAt的方法转为中文

2.实现自动补0的方法

3.通过字符串拼接 将本机电脑时间改为 2016年11月28日 星期一19时

55分18秒



14、Date应用

把时间格式的字符串变为我们的标准时间格式

- `var tar = new Date('2016-10-17 22:00:00');`
- `var tar = new Date('2016/10/17 22:00:00');`
- 注意：中间用-在IE6~8不兼容

