# HTML

1. 文字标签：修改文字的样式

<font></font>

属性：size：文字的大小。取值1-7，字体大小依次增大。若超出了7，默认还是7。

color:文字颜色。取值为英文单词red、green等或16位进制(#66cc66等)表示。

1. 注释标签

Html的注释标签：<!--注释内容-->

1. 标题标签

<h1></h1> <h2></h2>…..<h6></h6>，从<h1>到<h6>，大小是依次变小，同时会自动换行

1. 水平线标签

<hr/>

属性：size：水平线的粗细，取值1-7，粗细依次增粗。

color:颜色

<hr size=”5” color=”blue”/>

1. 特殊字符

要在网页上显示特殊字符，需要转义

< &lt;

> &gt;

空格 &nbsp;

& &amp;

1. 列表标签

<dl>

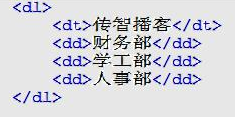
<dt></dt>

<dd></dd>

…(0个或多个<dd></dd>)

</dl>

如：



1. 有序列表标签

<ol></ol>

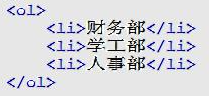
属性：type：设置排序方式。取值 1(默认) a i I A

<ol>

<li></li>

…(0个或多个<li></li>)

</ol> 如：



1. 无序列表标签

<ul></ul>

属性：type: 取值 (空心圆)circle、(默认实心圆)disc、(实心方块)square

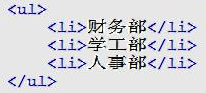
<ul>

<li></li>

…(0个或多个<li></li>)

</ul>

如：



1. 图像标签

<img src=”图片的路径”/>

属性：src：图片的路径

width：图片的宽度

height：图片的高度

alt：图片上显示的文字。把鼠标移动到图片上，停留片刻显示内容(兼容性差，了解即可)

1. 超链接标签

链接资源：

<a href=”链接到资源的路径”>显示在页面上的内容</a>

属性：href:链接的资源的地址

target：设置打开的方式，默认在当前页打开

取值：\_blank：在一个新窗口打开

\_self(默认): 在当前页面打开

当超链接不需要到任何地址时，最好在href里加#

<a href=”#”>链接</a>

定位资源：

如果想要定位资源，定义一个位置

<a name=”top”>顶部</a>

回到这个位置

<a href=”#top”>回到顶部</a>

1. 原样格式输出标签

<pre>内容</pre>

会将内容里的空格、换行等原样输出

1. 表格标签

<table></table>：表格的范围

属性：border:表格线

bordercolor:表格线的颜色

cellspacing:单元格之间的距离

width:表格的宽度

height:表格的高度

<tr></tr>

属性：align：取值：left、right、center

<td></td>

属性：align：取值：left、right、center

格式：<table>

<tr>

<th></th>

<td></td>

</tr>

</table>

定义一行使用<tr>

定义一个单元格使用<th>或<td>，二者可以混合使用，<th>的内容会加粗、居中

合并单元格：

rowspan：跨行。如<td rowspan=”2”>跨两行</td>

colspan：跨列。 如<td colspan=”2”>跨两列</td>

1. 表单标签

<form></form>：表单的范围

属性：action：提交到地址，默认提交到当前页面。

method：表单的提交方式，取值为get(默认)或者post。

大部分的输入项使用<input type=”类型”/>

需要指定name属性用来提交数据和标识标签。

普通输入项：<input type=”text”/>

密码输入项：<input type=”password”/>

单选输入项：<input type=”radio”/>

需要设置属性name

name的属性值必须相同

必须有一个value属性

实现默认选中的属性：checked=”checked”

复选输入项：<input type=”checkbox”/>

需要设置属性name

name的属性值必须相同

必须有一个value属性

实现默认选中的属性：checked=”checked”

文件输入项(上传的时候用到)：<input type=”file”/>

下拉输入项,如：

<select name=”birth”>

<option value=”1991”>1991</option>

<option value=”1992”>1992</option>

<option value=”1993”>1993</option>

</select>

实现默认选中的属性(<option>里的属性)：selected=”selected”

实现显示所有下拉项的属性(<select>里的属性):multiple=”multiple”

文本域，如：<textarea cols=”10” rows=”10”></textarea>

隐藏项(不会显示在页面上，但是存在于代码里)，其值会随表单一起提交：

<input type=”hidden”/>

提交按钮：

<input type=”submit” name=”submit” value=”注册”/>

图片按钮：

<input type=”image” src=”图片地址”/>

重置按钮：

<input type=”reset”/>

普通按钮(和js一起使用)：

<input type=”button” value=””/>

1. 其他常用标签

<b></b>内容加粗

<u></u>内容加上下划线

<i></i>内容倾斜

<sub></sub>下标

<sup></sup>上标

<p></p>段落标签

<div></div>框架标签，自动换行

<span></span>框架标签，不会自动换行

1. 头标签(了解)

在head里的标签就是头标签。如：<title></title>

<meta http-equiv=”refresh” content=”3;url=hello.html”/>

三秒后跳转到hello.html

<base target=”\_blank”>

统一设置页面里的超链接打开方式

link标签：用来引入其他文件

# CSS

1.css和html的结合方式(四种)

1. 在每个html标签上都有一个属性style，把css和html结合在一起

<div style=”background-color:red;color:green”>内容</div>

1. 使用html的一个标签实现，写在head里面

<style type=”text/css”>

div{

background-color:red;

color:green;

}

</style>

1. (一般不用)在style标签里面使用语句来引入css文件,如：

<style type=”text/css”>

@import url(div.css);

</style>

1. 使用头标签link，引入外部css文件

<link rel=”stylesheet” type=”text/css” href=”css文件路径”/>

四种方式的优先级(一般情况)：

由上到下，由外到内，优先级不断提高

2.css的基本选择器(三种)

(1)标签选择器

使用标签名作为选择器的名称,如：

div{

background-color:red;

color:green;

}

(2)class选择器

需要为html标签给予一个class属性，如：

<div class=”h”>aaaaaaa</div>

然后使用class选择器

.h{

background-color:red;

}

(3)id选择器

需要为html标签给予一个id属性，如：

<div id=”a”>aaaaaaa</div>

然后使用id选择器

#a{

background-color:red;

}

选择器的优先级：

<div style=”background-color:red;color:green”>内容</div>(这是style内部选择器)

style(内部)>id选择器>class选择器>标签选择器

3.css的扩展选择器

(1)关联选择器

如：设置div标签里p标签的样式(div嵌套p)

div p{

background-color:red;

}

(2)组合选择器

把div和p标签一起设置为相同的样式(没有嵌套),如：

div,p{

background-color:red;

}

(3)伪元素选择器(了解)

设置超链接的状态

原始状态 鼠标放上去的状态 点击状态 点击之后的状态

:link :hover :active :visited

如：

a:link{

background-color:black;

}

a { text-decoration:none;} 去除默认的A标签文本样式。

4.css的盒子模型

在进行布局前需要把数据封装到一块一块的区域内(div)

1. 边框

统一设置，格式：

border:大小 样式 颜色。如：

border:2px solid blue

分开设置，格式：

上border-top: 大小 样式 颜色

下border-bottom: 大小 样式 颜色。

左border-left: 大小 样式 颜色。

右border-right: 大小 样式 颜色。

1. 内边距

统一设置，格式：

padding：大小

分开设置，格式：

上padding -top: 大小

下padding -bottom: 大小

左padding -left: 大小

右padding -right: 大小

1. 外边距

统一设置，格式：

margin:大小

分开设置，格式

上margin -top: 大小

下margin -bottom: 大小

左margin -left: 大小

右margin-right: 大小

5.css的相关属性

(1)文字属性

font-style: 样式，如 italic

font-weight: 宽度，如 bold

font-size: 大小，如 10px

font-family: 字体，如 楷体

6.css布局的漂浮

float属性，取值：none(默认，不漂浮)、left(左漂浮)、right(右漂浮)

7.css布局的定位

position属性，取值：

absolute。

当取值为absolute时。此时被修饰的标签会飘起来，下面的标签**会**占据其原来位置。这时可使用top、bottom、left、right属性对标签进行定位。

relative

当取值为relative时。此时被修饰的标签会飘起来，下面的标签**不会**占据其原来位置。这时也可使用top、bottom、left、right属性对标签进行定位。

8.实用的css样式

(1) 可以使用 [background-color 属性](https://www.w3school.com.cn/cssref/pr_background-color.asp)为元素设置背景色。这个属性接受任何合法的颜色 值。如：p {background-color: gray;}

(2) 要把图像放入背景，需要使用 [background-image 属性](https://www.w3school.com.cn/cssref/pr_background-image.asp)。如：

body {background-image: url(/i/eg\_bg\_04.gif);}

(3) 如果需要在页面上对背景图像进行平铺，可以使用 [background-repeat 属性](https://www.w3school.com.cn/cssref/pr_background-repeat.asp)。

取值：repeat-x:图像只在水平方向平铺。repeat-y:图像只在垂直方向平铺。

no-repeat:不允许图像平铺。

1. 可以利用[background-position 属性](https://www.w3school.com.cn/cssref/pr_background-position.asp)改变图像在背景中的位置。

它的取值有三种，分别是关键字、百分比、长度值。

关键字有top、bottom、left、right 和 center。例如，top right 使图像放置在元素内边距区的右上角。根据规范，位置关键字可以按任何顺序出现，只要保证不超过两个关键字 - 一个对应水平方向，另一个对应垂直方向。如果只出现一个关键字，则认为另一个关键字是 center。

百分比值：如果你想把一个图像放在水平方向 2/3、垂直方向 1/3 处，可以这样声明：

background-position:66% 33%;如果只提供一个百分数值，所提供的这个值将用作水平值，垂直值将假设为 50%。

长度值：长度值解释的是元素内边距区左上角的偏移。偏移点是图像的左上角。

比如，如果设置值为 50px 100px，图像的左上角将在元素内边距区左上角向右 50 像素、向下 100 像素的位置上。background-position:50px 100px;

(5) 如果文档比较长，那么当文档向下滚动时，背景图像也会随之滚动。当文档滚动到超过图像的位置时，图像就会消失。你可以通过 [background-attachment 属性](https://www.w3school.com.cn/cssref/pr_background-attachment.asp)防止这种滚动。

取值：fixed(固定的)。background-attachment 属性的默认值是 scroll，也就是说，在默认的情况下，背景会随文档滚动。

9.禁用textarea标签的拖拽

设置此css样式resize:none;

# JS基础

1. js简介

js特点：

1. 交互性

信息的动态交互

1. 安全性

js不能访问本地磁盘的文件

1. 跨平台性

只要支持js的浏览器都能运行

js组成：

1. ECMAScript

由ECMA(欧洲计算机协会)制定的js语法、语句…..

1. BOM

broswer object model:浏览器对象模型

1. DOM

document object model:文档对象模型

1. js和html的结合方式(两种)

(1)内嵌式



(2)使用<script>标签直接写js代码

<script type=”text/javascript”>

js代码;

</script>

(3)使用<script>标签引入外部的js文件

<script type=”text/javascript” src=”js文件的路径”>

</script>

使用第三种方式时，不要在script标签内部写js代码了，写了也不会执行。

**除非你写了一个<script>标签负责引入外部.js文件，再写另外一个<script>标签用(2)**

**方式写js代码。这样外部文件会直接引入，同时另一个<script>标签内的代码也会**

**执行。如：**

<script type=”text/javascript” src=”js文件的路径”></script>

<script type=”text/javascript”>js代码;</script>

1. js的声明变量和原始类型

js中定义变量都使用关键字var

js的原始类型有5个

string(字符串)、number(数字)、boolean(true和false)、null(对象引用为空)、undifined(定义变量而没有赋值)

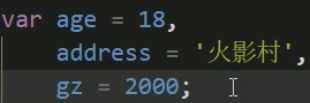
如果用boolean类型的值参与数学运算，则true看作1，false看作0.

null参与数学运算，看作0.

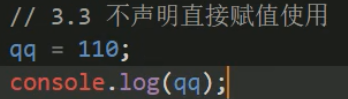
typeof(x)或者typeof x:查看当前变量x的数据类型

js变量的类型只有运行过程中才会被确认下来。

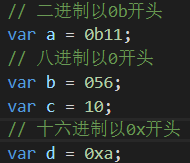
变量可以同时声明多个：

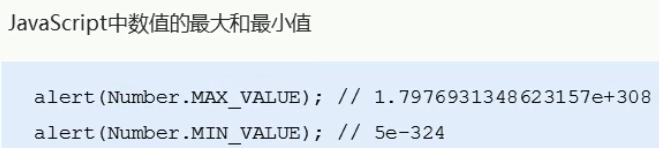


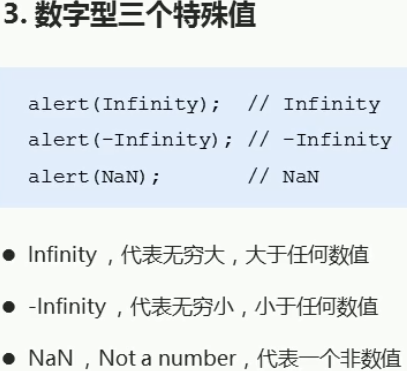
js的变量可以不声明直接使用，默认为全局变量：



js变量可以赋值二进制、八进制、十进制、十六进制。







1. js的语句

if判断语句、switch语句、循环语句(for、while、do-while)

1. js的运算符

js不区分整数和小数。

字符串相加仍是字符串；字符串相减是减法运算，若字符串不是数字会提示NaN错误。

boolean类型进行算术运算时，true为1，false为0。

==比较的只是值，而===比较的是类型和值。

document.write(“内容”); 将内容输出在页面上。内容可以是变量、常量、html代码。注意：双引号里不能再用双引号，用单引号代替

6.(补充内容)js的弹窗(常用于简单的调试和信息展示，如XSS漏洞的测试)

(1)警告弹窗alert(‘’)

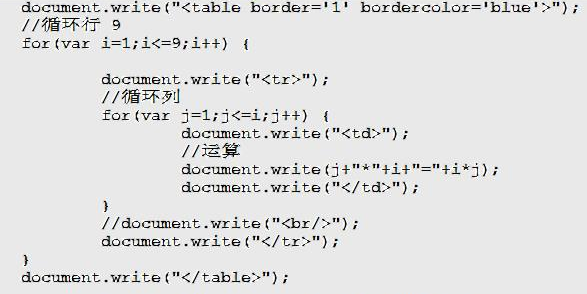
(2)确认弹窗confirm(‘’)

(3)提示输入弹窗prompt(‘’,‘’)或prompt(‘’)。

可以使用变量接收它得到的值：var myname = prompt(‘请输入：’);

接收的值会一律转化为字符串。

1. 实现99乘法表



1. js的数组

js数组的定义方式(三种)

(1)var arr = [1,2,3]; var arr = [1,”4”,true];

(2)使用内置对象Array

var arr = new Array(5);//定义一个长度为5的数组

arr[0] = “1”;

(3) 使用内置对象Array

var arr = new Array(3,4,5);//定义一个数组，数组元素是3,4,5

数组有一个属性 length：获取到数组的长度(数组元素的个数)。

数组可以存放不同的数据类型的数据。

1. js的函数

js里定义函数的方式(三种)

1. 使用关键字function

function 方法名(参数列表){

方法体；

返回值可有可无

}

1. 匿名函数

var add = function(参数列表){

方法体和返回值；

}

如：

var add = function(m,n){

alert(m+n);

}

//调用方法

add(5,6);

1. 使用js里的一个内置对象Function(了解)

var add = new Function(“参数列表”,”方法体和返回值”);

**补充：js的函数没有重载。**

1. js的全局变量和局部变量

全局变量：在script标签里，同时在方法外定义的变量。在页面中js部分都有效

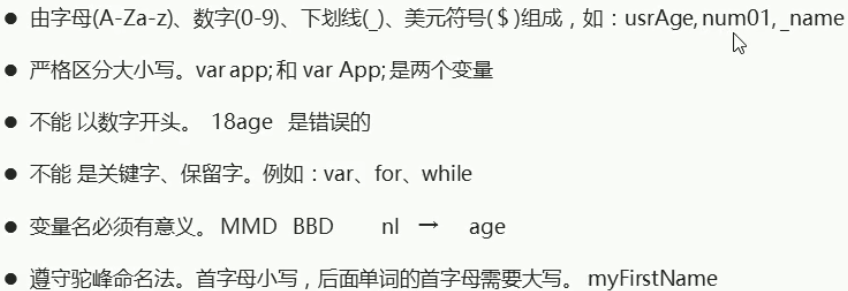
局部变量：在方法内部定义的变量，只能在其方法内部使用

补充：在IE8极其以上版本中，按F12会出现调试工具

1. script标签位置

由于html是从上到下解析的，所以建议script标签放在</body>后，保证js代码的执行

1. js变量的命名规范



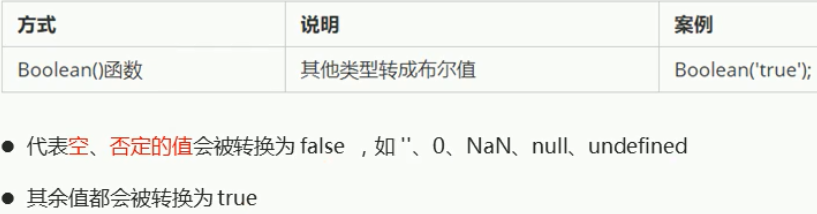
1. 数据类型转换。
2. 转换为字符串



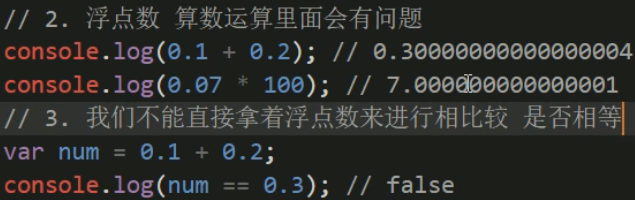
1. 转换为数字型



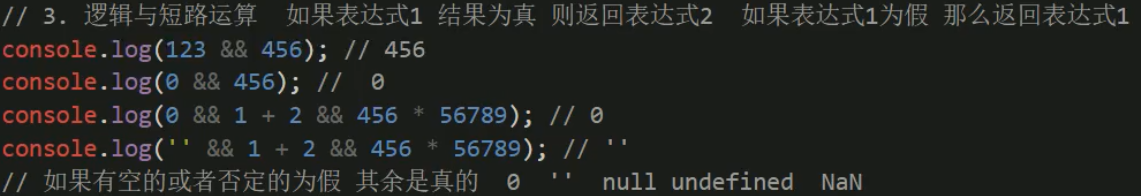
1. 转换为布尔型



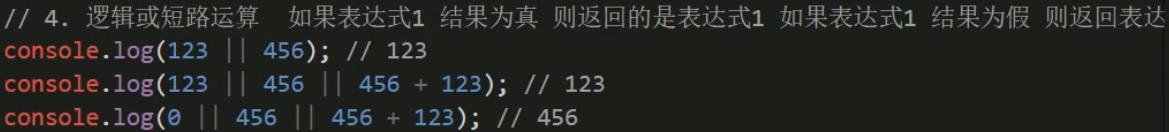
1. 算数运算符



1. 逻辑与的逻辑中断



15. 逻辑或的逻辑中断



16.运算符优先级



# JS对象

1. js的String对象

创建String对象

var str = “abc”;

属性

length:字符串的长度

方法

bold():加粗

fontcolor():设置字符串的颜色

fontsize():设置字体的大小

link():将字符串显示成超链接

str.link(“hello.html”);

sub()和sup():设置为下标和上标

concat():连接字符串

var str1="Hello ";

var str2="world!";

document.write(str1.concat(str2));

charAt():返回指定位置的字符

var str="Hello world!";

document.write(str.charAt(1));

indexOf():返回字符串位置,若字符串不存在，返回-1

var str="Hello world!";

document.write(str.indexOf("world"));

split():切分字符串，成数组

var str="How are you doing today?";

document.write(str.split(" ") + "<br />");

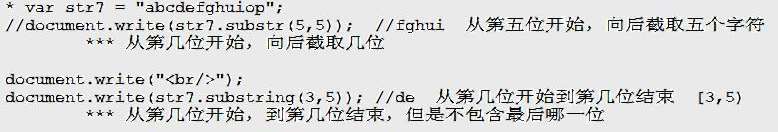
replace():替换字符串。**可以和正则表达式配合使用**

需要传递两个参数，第一个是原始字符串，第二个是要替换成的字符串

var str = “abcde”;

document.write(str.replace(“a”,”A”));

substr()和substring()

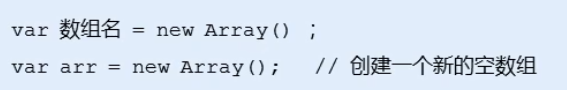


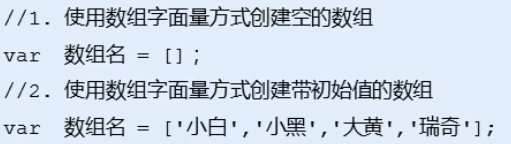
转义字符



字符串用+号与其他类型的变量拼接，其结果都是字符串。

1. js的Array对象





数组中可以存放任意类型。

数组名.length 可以得到数组的长度。

可以通过修改数组名.length的值来修改数组的长度。

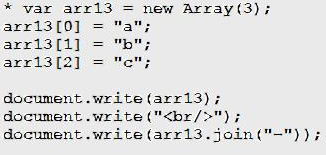
不要直接给数组名赋值，否则里面之前的元素都会消失。

方法：

concat():数组的连接



join():根据指定的字符分割数组，返回一个字符串

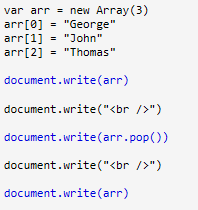


push():向数组末尾添加元素，返回数组的新长度(实参数量可以是多个)



unshift():向数组最前面添加元素，返回数组的新长度(实参数量可以是多个)

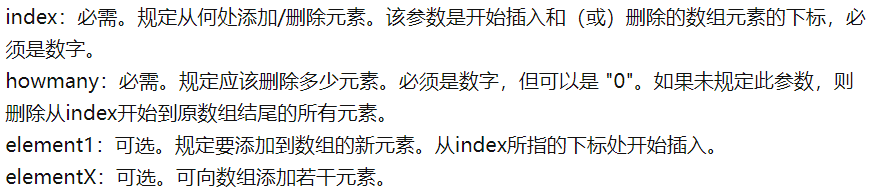
pop():删除最后一个元素，返回删除的那个元素



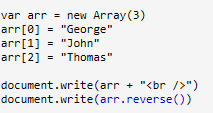
shift():删除第一个元素，返回删除的那个元素

splice(index,howmany,element1,.....,elementX):

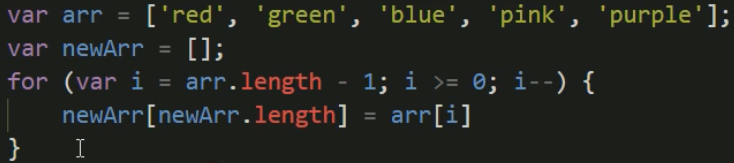
用于插入、删除或替换数组的元素,返回被删除元素组成的数组

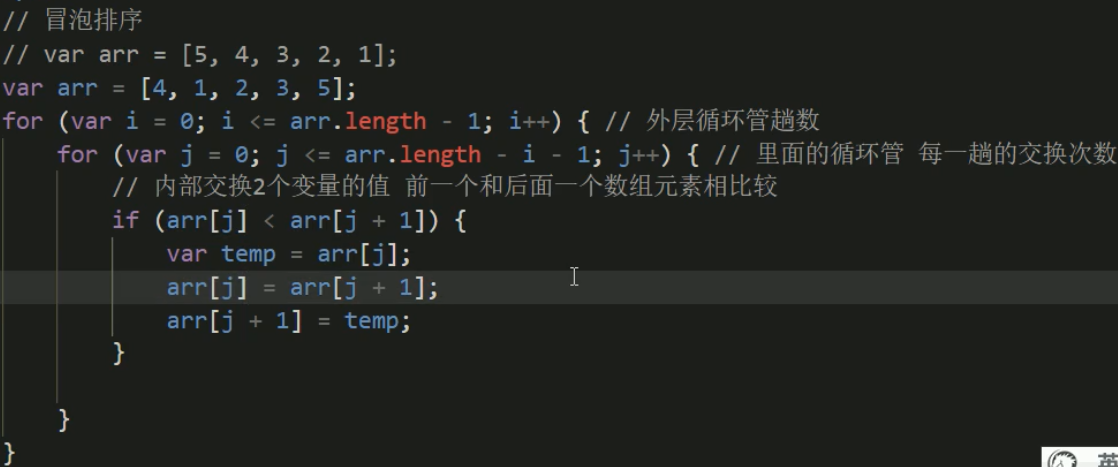


reverse():反转数组中的元素



反转的底层实现：

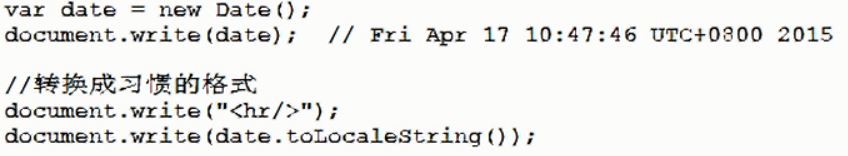




1. js的Date对象

var date = new Date();

获取当前时间



获取当前的年

getFullYear()



获取当前的月(0-11)

getMonth()



获取当前的日(1-31)

getDate()



获取当前的星期(0-6)

getDay()



获取当前的小时

getHours()

获取当前的分钟

getMinutes()

获取当前的秒

getSeconds()

获取当前的毫秒(1970年1月1日至今的毫秒值)

getTime()

1. js的Math对象

Math对象的方法都是静态方法，可以直接Math.方法名()调用

ceil(x):向上舍入

floor(x):向下舍入

round(x):四舍五入

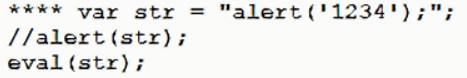
random():得到随机数，范围[ 0,1)。得到[min,max)范围内的随机数：

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

1. js的全局函数

由于不属于任何一个对象，直接写名称使用

eval():执行js代码



encodeURL():对字符进行编码

decodeURL():对字符进行解码

encodeURIComponent() :可把字符串作为 URI 组件进行编码

[decodeURIComponent()](mk:@MSITStore:F:\Document\W3School离线手册(2017.03.11版).chm::/www.w3school.com.cn/jsref/jsref_decodeuricomponent.asp):解码一个编码的 URI 组件

isNaN(x):判断当前字符串是否是数字

**如果是数字，返回false，不是数字返回true**

parseInt():将字符串转换为整数

1. js方法的重载

**js中的方法不存在重载，但可以通过arguments数组模拟重载**

1. js的bom对象

bom：broswer object model：浏览器对象模型

有哪些对象：

navigator：获取客户浏览器的信息

navigator.appName;

screen:获取屏幕的信息

screen.width;

screen.height;

location:请求url地址

location.href;//获取请求的url地址

location.href = “hello.html”;//跳转到hello.html页面

history:请求的url历史纪录

history.back()或history.go(-1):到访问的上一个页面

history.forward()或history.go(1):到访问的下一个页面

window:窗口对象，也是顶层对象(所用的bom对象都是在window里面操作的)

方法

window.alert(“显示的内容”):页面弹出一个框，显示内容。简写为alert(“显示的内容”)

confirm():确认框,返回一个boolean值。

var flag = window.confirm(“显示的内容”);

prompt(“text”,”value”):输入框。text为提示内容，value为默认值

window.prompt(“please input:”,”0”);

open():打开一个新的窗口



close():关闭窗口。(兼容性差)

setInterval(“js代码”,毫秒值)

//每三秒执行一次alert方法，同时返回一个id值

window.setInterval(“alert(‘123’)”,3000);

setTimeout(“js代码”,毫秒值)

//三秒后执行alert方法，只执行一次，同时返回一个id值

window. setTimeout (“alert(‘123’)”,3000);

clearInterval(id值):通过id值清除setInterval设置的定时器

clearTimeout (id值):通过id值清除setTimeout设置的定时器

1. js的dom对象

dom：document object model：文档对象模型

1. document对象，表示整个的文档

常用方法

write():用来输出字符串或html代码。

document.write(“111”);

document.write(“<br/>”);

getElementById(“id名称”):通过id名称返回一个标签对象。例子：

<input type=”text” id=”name” name=”a”/>

<script type=”text/javascript”>

//使用getElementById得到input标签

var in= document. getElementById(“name”);

//弹出input标签里面的name值

alert(in.name);

//设置input标签的value值

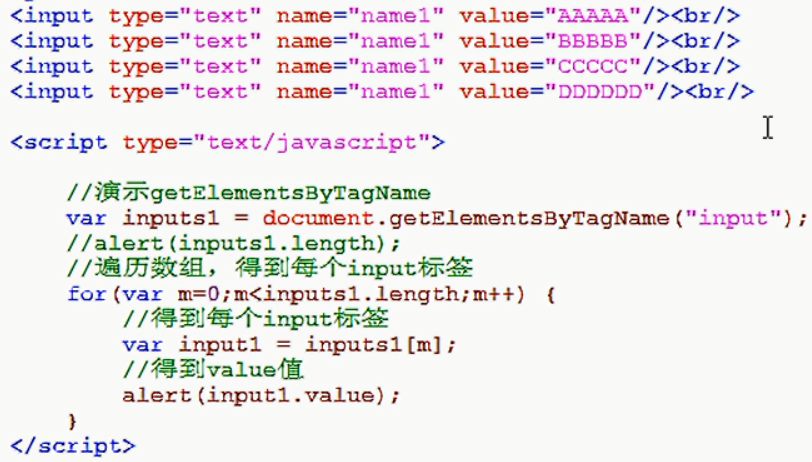
in.value = “bbb”;

</script>

getElementsByName(“name值”):通过标签的name属性返回标签集合(数组)。例子：



getElementsByTagName(“标签名”):通过标签名返回标签集合(数组)。例子：



注意:当input标签只有一个且你想用getElementsByName()或getElementsByTagName()获取对象时,可以这样:

<input type=”text” name=”na” value=”AAA”/>

<script type=”text/javascript”>

var in = document.getElementsByName(“na”)[0];

var in2 = document.getElementsByTagName(“input”)[0];

alert(in.value);

alert(in2.value);

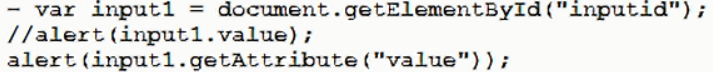
</script>

10.元素对象(element对象)

要操作element对象，首先要获取element，使用getElementById(“id名称”)或getElementsByName(“name值”)或getElementsByTagName(“标签名”)这三个方法可以获取element

方法

getAttribute(“属性名称”):获取属性里的值,例:



setAttribute(“属性名称”,”值”):设置属性里的值，例：

input1.setAttribute(“class”,”haha”);

removeAttribute(“属性名称”):删除属性里的值,但无法删除value里的值，例：

input1.removeAttribute(“name”);

1. Node对象属性

nodeName、nodeType、nodeValue

使用dom解析html时候，需要里面的标签、属性和文本都封装成对象

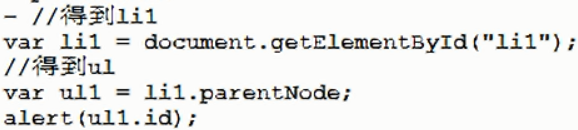


1. Node对象的属性二

父节点

ul是li的父节点

可以通过对象的属性parentNode获取其父节点对象

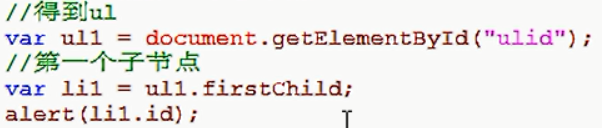


子节点

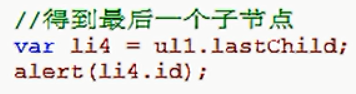
li是ul的子节点

childNode：获取所有子节点对象，但是兼容性很差

firstChild：获取第一个子节点对象



lastChild：获取最后一个子节点对象

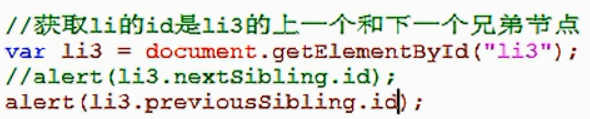


同辈节点

li与li之间是同辈节点

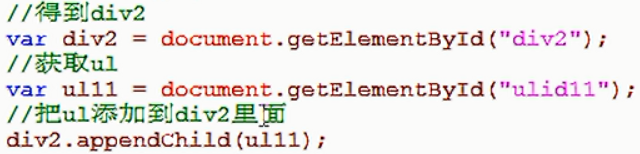
nextSibling：返回一个节点的下一个同辈节点对象

previousSibling：返回一个节点的上一个同辈节点对象



1. 操作dom树

appendChild(子节点对象):添加子节点到末尾，实现类似于剪切+粘贴的效果



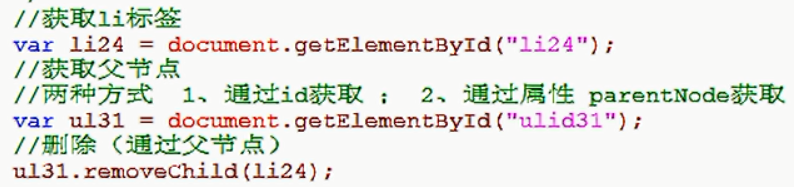
insertBefore(newNode,oldNode):在某个节点之前插入一个新的节点

newNode：要插入的节点对象(应该先创建)

oldNode：在哪个节点对象之前插入



removeChild(子节点对象):删除子节点



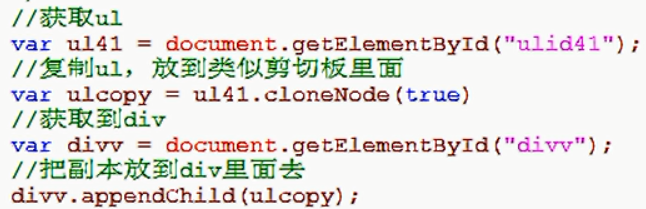
replaceChild(newNode,oldNode):替换节点

newNode：新的节点对象

oldNode：旧的节点对象(被替换)



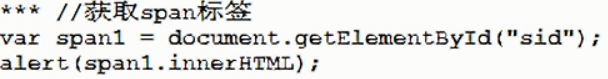
cloneNode(boolean):复制节点



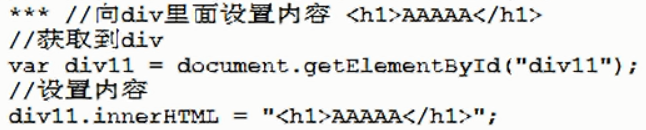
1. innerHTML属性

这个属性不是dom的组成部分，但大多数浏览器都支持的属性

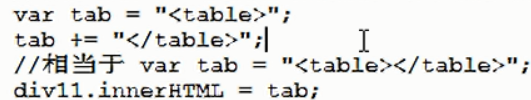
作用1：获取文本内容



作用2：向标签里设置内容(可以是html代码)



向div里面添加一个表格：



1. 案例：动态显示时间



1. 复选框和下拉列表的js属性

在复选框中，有一个属性checked表示该复选框是否被选中

<input type=”checkbox” id=”a” name=”love”/>篮球

<input type=”checkbox” id=”b” name=”love”/>排球

<script type=”text/javascript”>

var a = document.getElementById(“a”);

//选中篮球复选框

a.checked = true;

//取消选中篮球复选框

a.checked = false;

</script>

在下拉列表中，有一个属性selected表示该选项是否被选中

<select>

<option id=”op” value=”1945”>1945</option>

<option value=”1946”>1946</option>

</select>

<script type=”text/javascript”>

var op = document.getElementById(“op”);

//选中1945选择框

op.selected = true;

//取消选中1945选择框

op.selected = false;

</script>

16.其他html提交方式及事件

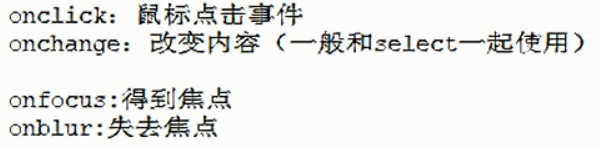
type=button的提交方式



超链接提交方式



事件



17.原生页面加载函数

window.onload = function(){

alert(‘页面加载，就会执行的函数’);

}