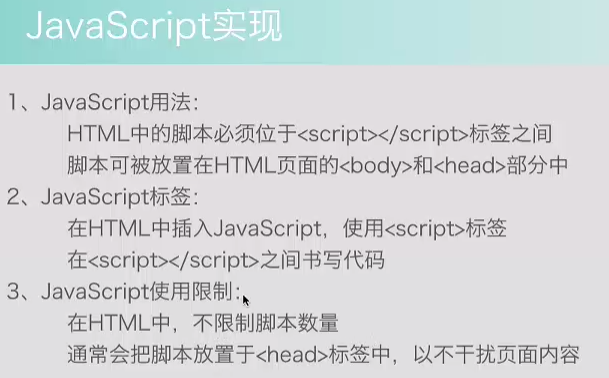
# JavaScript基础

# Javascript基础教程



## 基本输出语句:

document.write(“Hello JS”);//打印树池

Document.write(“<h1>标签H1</h>”);//输出内容可以添加HTML5标签

### 通过JS脚本改变其他标签内容样式

<p id="pid">HELLO</p>

<script>

document.getElementById("pid").innerHTML="JavaScript";//通过脚本改变p标签内容

</script>

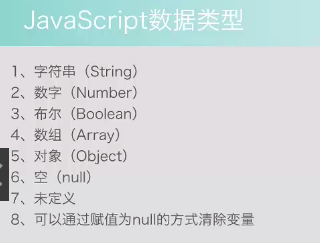
## 语法和注释

执行顺序按照顺序执行编写

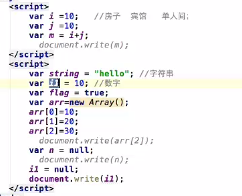
单行注释: //

多行注释: /\*\*/

## 变量和数据类型



Var string = “字符串”;



# JavaScript函数

## 函数用途

### Alert();弹出对话框

<script>

var a=10;

var b=20;

var sum=a+b;

alert(sum);//弹出对话框显示结果

function demo(a,b){

var sum = a+b;

return sum;

}

var v1 = demo(10,20);

alert(v1);//创建方法并调用

</script>

## 定义函数

Fucation 函数名(){

函数体; (代码块)

}

## 调用函数

### 调用方式:

#### 在<script>标签内调用

Fucation demo(){

Var a =1;

Var b =2;

Var sum =a+b;

Alert(sum);

}//定义函数

Demo();//调用函数

2.在HTML文件中调用

<input type=”button” value=”按钮” onclick=”demo()”>

<button onclick=”demo()”>按钮</button>

//调用demo方法

## 带参数的函数

例如: demo(arg1,arg2);

Fucation demo(a,b){

Var sum =a+b;

Alert(sum);

}

调用:demo(10,20);

<button onclick=”demo(10,20)”>按钮</button>

## JavaScript异常处理和事件处理

### 异常捕获

Try{

发生异常的代码块

}catch(err){

错误信息处理;

}

<script>//异常捕获

function demo(){

try{

alert(str);

}

catch(err){

alert(err);

}

}

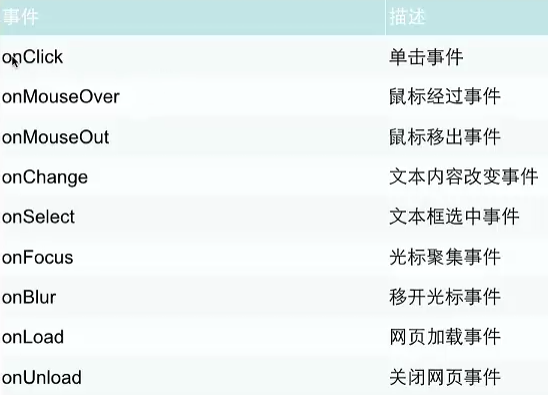
demo();

</script>

通过异常捕获判断输入是否为空



## JavaScript事件



<button onClick="demo()">按钮</button><!--通过鼠标单间按钮事件调用方法demo-->

<script>

function demo(){

alert("Hello");

}

</script>

鼠标经过\移除事件

<div class="div" onMouseOut="onOut(this)" onMouseOver="onOver(this)" style="width: 200px ; height: 200px; background-color: antiquewhite"></div>

<script>

function onOut(ooj){

ooj.innerHTML="Hello";

}

function onOver(ooj){

ooj.innerHTML="World";

}

</script>

## JS HTML DOM (文档对象模型)对象

### DOM简介

通过可编程的对象模型，JavaScript 获得了足够的能力来创建动态的 HTML。

JavaScript 能够改变页面中的所有 HTML 元素

JavaScript 能够改变页面中的所有 HTML 属性

JavaScript 能够改变页面中的所有 CSS 样式

JavaScript 能够对页面中的所有事件做出反应

### 查找Html元素

#### 通过id查找HTML元素

本例查找id = “intro” 元素

Var x = document.getElementById(“intro”);

如果找到该元素，则该方法将以对象（在 x 中）的形式返回该元素。

如果未找到该元素，则 x 将包含 null。

#### 通过标签名查找HTMl元素

本例查找 id="main" 的元素，然后查找 "main" 中的所有 <p> 元素：

Var x = document.getElementById(“main);

Var y = x.getElementsByTagName(“p”);

### DOM HTML

HTML DOM 允许 JavaScript 改变 HTML 元素的内容。

#### 改变HTML输出流

<script>

Docunment.write(Date());//输出当前日期

</script>

#### 改变HTML内容

修改 HTML 内容的最简单的方法时使用 innerHTML 属性

Document.getElementById(id).innerHTML = new HTMl

<script>

Document.getElementById(“p1”).innerHTML=”New text”;

</script>

<script>

Var element = document.getElementById(“header”);

Element.innerHTML = “”New Header;

</script>

#### 改变HTMl属性

<img id="image" src="smiley.gif">

<script>

document.getElementById("image").src="landscape.jpg";

</script>

### DOM CSS

### HTML DOM 允许 JavaScript 改变 HTML 元素的样式。

#### 改变HTML样式

语法:

Document.getElementById(id).style.property/\*样式\*/=new style;

例:改变p2颜色

<script>

Document.getElementById(“p2”).style.color=”blue”;

</script>

例2:

<h1 id="id1">My Heading 1</h1>

<button type="button" onclick="document.getElementById('id1').style.color='red'">

点击这里

</button>

### JavaScript HTML DOM 事件

HTML DOM 使 JavaScript 有能力对 HTML 事件做出反应

<h1 onclick="this.innerHTML='谢谢!'">请点击该文本</h1>

onload 和 onunload 事件

onload 和 onunload 事件会在用户进入或离开页面时被触发。

onload 事件可用于检测访问者的浏览器类型和浏览器版本，并基于这些信息来加载网页的正确版本。

onload 和 onunload 事件可用于处理 cookie。

<body onload="checkCookies()">//页面载入时清除cookie

### Script HTML DOM 元素（节点）

#### 创建新的HTML元素append

如需向 HTML DOM 添加新元素，您必须首先创建该元素（元素节点），然后向一个已存在的元素追加该元素。

<div id="div1">

<p id="p1">这是一个段落</p>

<p id="p2">这是另一个段落</p>

</div>

<script>

var para=document.createElement("p");

//创建新的p元素

var node=document.createTextNode("这是新段落。");

//创建文本节点,向p元素中添加文本

para.appendChild(node);

//向p元素中追加这个文本节点

var element=document.getElementById("div1");

element.appendChild(para);

//找到一个已有元素,向这个元素中追加新元素para

</script>

#### 删除已有元素remove

如需删除 HTML 元素，您必须首先获得该元素的父元素：

<div id="div1">

<p id="p1">这是一个段落。</p>

<p id="p2">这是另一个段落。</p>

</div>

<script>

var parent=document.getElementById("div1");

var child=document.getElementById("p1");

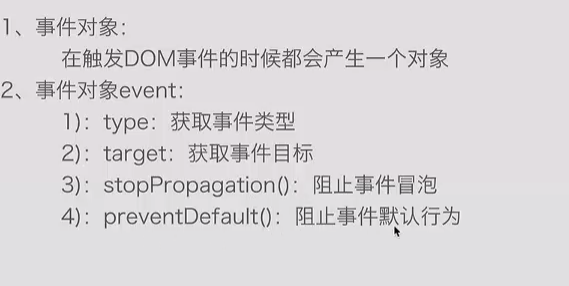
parent.removeChild(child);

</script>

## Js事件详解

## 

## Js事件对象



Alert(event.type);

## JavaScript Windows

### Window location

window.location 对象用于获得当前页面的地址 (URL)，并把浏览器重定向到新的页面。

* location.hostname 返回 web 主机的域名
* location.pathname 返回当前页面的路径和文件名
* location.port 返回 web 主机的端口 （80 或 443）
* location.protocol 返回所使用的 web 协议（http:// 或 https://）
* location.assign() 方法加载新的文档。

## JS popupalert(消息框)

<div id="popupalert">JS消息框</div>

<button onClick="alert('这是一个警告框'+'\n'+'警告框！')">警告框</button>

<button onClick="confirm1()">确认框</button>

<button onClick="prompt1()">提示框</button>

<script>

function confirm1(){

var v=confirm("这是确认框，有两个返回值true/false,请选择一个");

if(v==true){

alert("你按了确认");

}

else{

alert("你按了取消");

}

}

</script>

<script>

function prompt1(){

var name= prompt("请输入你的姓名：","(例如:Tom)");//prompt('这是提示框（文本）','默认值（输入框值显示）')

if(name!=null && name!=""){

alert("你好："+name+"同学");

}

}

## Js timing(计时器)

**setTimeout()**

未来的某时执行代码

**clearTimeout()**

取消setTimeout()

function startTime()

{

var today=new Date()

var h=today.getHours()

var m=today.getMinutes()

var s=today.getSeconds()

// add a zero in front of numbers<10

m=checkTime(m)

s=checkTime(s)

document.getElementById('txt').innerHTML=h+":"+m+":"+s

t=setTimeout('startTime()',500)

}

function checkTime(i)

{

if (i<10)

{i="0" + i}

return i;

}

## JavaScript cookies

什么是cookie?

cookie 是存储于访问者的计算机中的变量。每当同一台计算机通过浏览器请求某个页面时，就会发送这个 cookie。你可以使用 JavaScript 来创建和取回 cookie 的值。

# JavaScript对象实例(代码加)

<!doctype html>

<html>

<head>

<meta charset="utf-8">

<title>JavaScript\_Model</title>

</head>

<body onLoad="message()">

<!--Basics-->

<script>

document.write("Hello");//页面载入时输出一个文本

document.write("<div><h1 style='color:red'>标题1</h1></div>");//载入时生成普通文本和标签

function message(){

alert("该提示窗口body载入时调用");//简单方法。弹出窗口

}

var name;//声明一个变量

name="张三";//为这个变量赋值

document.write(name);//使用这个变量

document.write("<br/>");

function countDown(){

var t= new Date();

var now = t.getHours();

if(now>10){

document.write("<b>早安</b>");

}

}

</script>

<!--JS函数-->

<script>

function myFunction(txt){//生命一个带参数的函数

alert(txt);

}

</script>

<form>

<input onClick="myFunction('您好!')" type="button" value="调用函数"><!--调用函数-->

</form>

<!--JS循环-->

<!--FOR循环-->

<script>

document.write("循环："+"<br/>");

for(i=0;i<5;i++){//声明一个for循环，无需声明被循环值的类型

document.write("第"+(i+1)+"次循环！")

document.write("<br/>");

document.write("<h"+(i+1)+">"+"标题"+(i+1)+"</h"+(i+1)+">");

}

</script>

<!--while循环-->

<script>

i= 0;

while(i<5){

document.write("第"+(i+1)+"次循环！"+"<br/>");

i++;

}

</script>

<!--使用for——IN遍历数组内元素-->

<script>

var x

var arr=new Array()

arr[0]="数组1"

arr[1]="数组2"

arr[2]="数组3"

for(x in arr){//for X in Array

document.write(arr[x]+"<br/>");

}

</script>

<!--throw声明-->

<script>

var x = prompt("请输入1-10之间的数字：","");//定义一个输入框（text，默认值）

try{

if(x>10){

throw "Err1"}

else if(x<0){

throw "Err2"}

else if(isNaN(x)){//输入的不是数字

throw "Err3"}

}

catch(er){

if(er=="Err1")

alert("该值过大");

if(er=="Err2")

alert("该值过小");

if( er=="Err3")

alert("请输入数字");

}

</script>

<!--onerrpr事件-->

<script>

onerror=handleErr

var txt=""

function handleErr(msg,url,l)

{

txt="本页中存在错误。\n\n"

txt+="错误：" + msg + "\n"

txt+="URL: " + url + "\n"

txt+="行：" + l + "\n\n"

txt+="点击“确定”继续。\n\n"

alert(txt)

return true

}

function message2()

{

adddlert("Welcome guest!")

}

</script>

<input type="button" value="查看消息" onclick="message2()" />

<!--为字符串添加样式-->

<script>

var txt="HeloWorld";

document.write("<p>Big: " + txt.big() + "</p>")

document.write("<p>Small: " + txt.small() + "</p>")

document.write("<p>Bold: " + txt.bold() + "</p>")

document.write("<p>Italic: " + txt.italics() + "</p>")

document.write("<p>Blink: " + txt.blink() + " (does not work in IE)</p>")

document.write("<p>Fixed: " + txt.fixed() + "</p>")

document.write("<p>Strike: " + txt.strike() + "</p>")

document.write("<p>Fontcolor: " + txt.fontcolor("Red") + "</p>")

document.write("<p>Fontsize: " + txt.fontsize(16) + "</p>")

document.write("<p>Lowercase: " + txt.toLowerCase() + "</p>")

document.write("<p>Uppercase: " + txt.toUpperCase() + "</p>")

document.write("<p>Subscript: " + txt.sub() + "</p>")

document.write("<p>Superscript: " + txt.sup() + "</p>")

document.write("<p>Link: " + txt.link("http://www.w3school.com.cn") + "</p>")

document.write(txt.indexOf("world")+"<br/>")//返回字符串中指定文本首次出现的位置 - indexOf()方法

document.write(txt.match("world"))//查找字符串中特定的字符，若找到，则返回该字符 - match() 方法

document.write(txt.replace(/Microsoft/,"W3School"))//查找字符串中特定的字符，若找到，则返回该字符 - match() 方法(被替换文本，替换后文本)

</script>

<!--JS Date对象实例-->

<script>

document.write("<br/><br/><br/>"+Date());//返回当前日期

var t =new Date()

document.write("<br/>从1970年到现在已经有"+t.getTime()+"ms了")

/\*用getDay来显示日期\*/

var weekday=new Array(7)

weekday[0]="星期日"

weekday[1]="星期一"

weekday[2]="星期二"

weekday[3]="星期三"

weekday[4]="星期四"

weekday[5]="星期五"

weekday[6]="星期六"

document.write("<br/>今天是" + weekday[t.getDay()]+"<br/>")

</script>

<!--JS Array数组实例-->

<!--将两个数组合并——concat（）-->

<script>

var arr = new Array(3)

arr[0] = "George"

arr[1] = "John"

arr[2] = "Thomas"

var arr2 = new Array(3)

arr2[0] = "James"

arr2[1] = "Adrew"

arr2[2] = "Martin"

document.write(arr.concat(arr2)+"<br/>")

//用数组的元素组成字符串 - join()

document.write(arr.join()) //将数组元素组成字符串并打印

document.write("<br/>")

document.write(arr.join("."))

document.write("<br/>")

document.write(arr.join("!"))

//文字数组-sort（）；

document.write("<br/>"+"文字数组："+arr.sort())//显示为排序后的数组

//数字数组-sort（）；

function sortNumber(a, b)

{

return a - b//方法 升序，如降序，把a-b改成b-a

}

var arr3 = new Array(6)

arr3[0] = "10"

arr3[1] = "5"

arr3[2] = "40"

arr3[3] = "25"

arr3[4] = "1000"

arr3[5] = "1"

document.write(arr3+"<br/>")

document.write(arr3.sort(sortNumber))//显示排序后的数组

</script>

<!--JS Boolean(布尔)对象实例-->

<script>

var b1=new Boolean( 0)

var b2=new Boolean(1)

var b3=new Boolean("")

var b4=new Boolean(null)

var b5=new Boolean(NaN)

var b6=new Boolean("false")

document.write("0 是逻辑的 "+ b1 +"<br />")

document.write("1 是逻辑的 "+ b2 +"<br />")

document.write("空字符串是逻辑的 "+ b3 + "<br />")

document.write("null 是逻辑的 "+ b4+ "<br />")

document.write("NaN 是逻辑的 "+ b5 +"<br />")

document.write("字符串 'false' 是逻辑的 "+ b6 +"<br />")

</script>

<!--JS Math(算术对象)实例-->

<!--使用 round() 对数字进行舍入-->

<script>

document.write(Math.round(0.60) + "<br />")

document.write(Math.round(0.50) + "<br />")

document.write(Math.round(0.49) + "<br />")

document.write(Math.round(-4.40) + "<br />")

//四舍五入

//使用 random() 来返回 0 到 1 之间的随机数

document.write(Math.random())

//使用 max() 来返回两个给定的数中的较大的数

document.write("<br/>");

document.write(Math.max(1,10))

//使用 min() 来返回两个给定的数中的较小的数

document.write("<br/>");

document.write(Math.min(-2,10))

</script>

</body>

</html>

# JavaScript框架(库)

## jQuery

如需在您的网页中使用 JavaScript 框架库，只需在 <script> 标签中引用该库即可：

引用 jQuery

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js">

</script>

## Prototype

## MooTools