Date 2017/10/16

HTML5开发开发知识学习笔记

HTML5基础

# HTML5元素、属性和格式化

几种标签：<a href=”链接地址”>显示内容</a>

<img src=”图像地址”></img>

<p></p>段落标签

<b></b>定义粗体文本

<em></em>定义着重文字

<i></i>定义斜体字

<sub></sub>定义下标

<sup></sup>定义上标

<ins></ins>定义插入字

<del></del>定义删除字

<br/>换行标签（闭合标签 空元素 ，文字后使用）

<hr/>创建水平线

## <meta>元数据，提供关于 HTML 文档的元数据。元数据不会显示在页面上，但是对于机器是可读的

## 一些搜索引擎会利用 meta 元素的 name 和 content 属性来索引您的页面。

## <meta name=”关键词” content=”关键词、关键词、关键词”>

# HTML5属性使用方法

属性以键/值对的形式出现

如:href=”www.abcd.com”

常用标签属性:

<h1>：align对齐方式（<h1 align=”center”>文本内容</h1>）

<body>:bgcolor 背景颜色

<a>:target 规定在何处打开连接（\_blank:从新标签打开链接内容）

通用属性：

class：规定元素的类名

id ：规定元素唯一ID

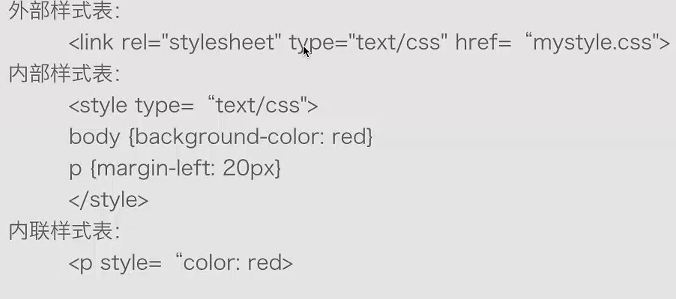
style：规定元素样式（头文件使用<style type=”text/css”></style>）

link ：资源引用

title ：规定元素额外信息（头文件使用）

# HTML5样式、链接和表格

三种样式表的插入方法：



HTML连接属性及使用

<a href=”地址”><img href=”图片地址” alt=”介绍图片内容”></img></a>(创建图片链接 ，图片失效时显示alt内容)

name属性：创建文档内的链接（<a name=”tips”>hello</a> <a herf=”#tips”>跳转到hello</a>）

HTML表格



# HTML5列表、块和布局

# HTML列表

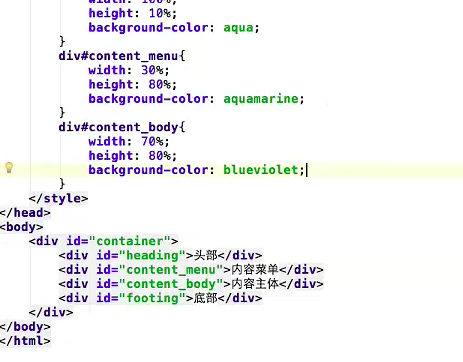
# 

# HTML块

# HTML5注释（<!-- 注释内容 -->command+/）

# HTML5布局

<div>元素布局



<table>元素布局

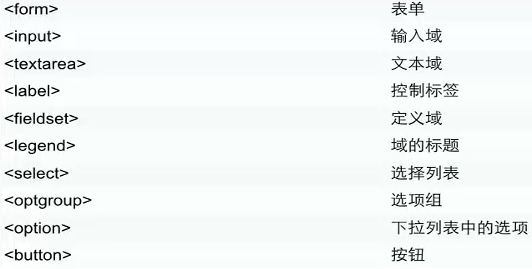


# HTML5表单提交和PHP环境搭建

# HTML表单

<表单常用于获取不同类型的用户输入

常用的表单标签



单选按钮：例如

请选择你的性别：男<form> <input type=”redio” name=”sex”>女<input type=”redio” name=”sex”></form>



(使用时name属性为将男、女分为一组，不使用会变成复选框效果)

复选框CheckBox

创建按钮 button，submit（默认提交按钮）

下拉列表<select> <option value=”1” selected>内容1</option><option>内容2</option></select>

文本域<textarea name=”text” cols=”30” rows=”30”>加入内容</textarea>（cols:列 rows:行）

样例：





# HTML表单与PHP交互

# 

# action：服务器链接地址

# method：获取方式（分为get、post（适用于密码、更安全））

# PHP内容

# 

# 

# HTML5框架、背景和实体容

## Html5框架

<html>

<frameset cols="25%,50%,25%">

<frame src="/example/html/frame\_a.html">

<frame src="/example/html/frame\_b.html">

<frame src="/example/html/frame\_c.html">

<noframes>

<body>您的浏览器无法处理框架！</body>

</noframes>

</frameset>

</html>

# 

### Iframe删除边框

<iframe src=”url” fromeboder=”0”></iframe>

### 使用iframe作为链接目标

<iframe src=”url” name=”iframe\_a”></iframe>

<a href=”url” target=”iframe\_a”></a>

## HTML5 Web Stroage概述

## 

sessionStorage示例

## 

## 

## localstroage示例

## 

## 

# html5本地数据库

数据库的使用

1. 创建数据库对象
2. 使用事务处理

<script>

/\*Var db = openDatabase(“数据库名称”，“数据库版本“，”数据库描述“，”数据库大小“)；\*/

Var db = openDatabase(“mydb”,”1.0”,”test db”,1024\*100);//创建数据库对象

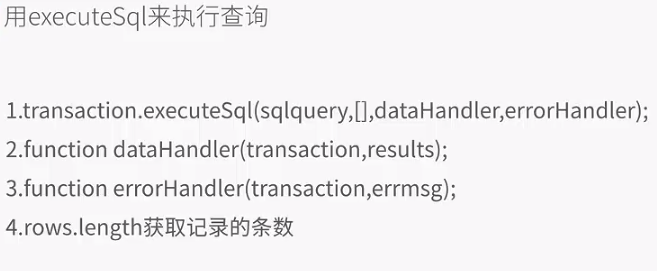
Db.transaction(fucation(tx){//使用事务处理

Tx.executeSql(“”);//执行查询

}）

</script>

1. 通过executeSql来执行查询



<script>

/\*Var db = openDatabase(“数据库名称”，“数据库版本“，”数据库描述“，”数据库大小“)；\*/

Var db = openDatabase(“mydb”,”1.0”,”test db”,1024\*100);//创建数据库对象

Db.transaction(fucation(tx){//使用事务处理

Tx.executeSql(“”);//执行查询

}）

Transaction.exesuteSql(“参数1”,[2],funcation(){};funcation(){});

//参数1：查询、修改、删除等语句

//2:数组

//3:成功时使用的回掉函数

//4.失败时使用的回掉函数

//rows属性 rows.length rows[index]

</script>

CSS基础

# CSS层叠样式表

## 样式表的三种使用方法

### 外部样式表

<head>

<link rel="stylesheet" type="text/css" href="mystyle.css">

</head>

### 内部样式表

<head>

<style type="text/css">

body {background-color: red}

p {margin-left: 20px}

</style>

</head>

### 内联样式

<head>

<style type="text/css">

body {background-color: red}

p {margin-left: 20px}

</style>

</head>

## css文本

## C:\Users\MIngdeng\Pictures\web课程截图\捕获.PNG

## Css字体

## C:\Users\MIngdeng\Pictures\web课程截图\捕获2.PNG

## Css链接a的四种状态

## **C:\Users\MIngdeng\Pictures\web课程截图\捕获3.PNG**

## Css列表

## C:\Users\MIngdeng\Pictures\web课程截图\捕获4.PNG

## C:\Users\MIngdeng\Pictures\web课程截图\捕获5.PNG

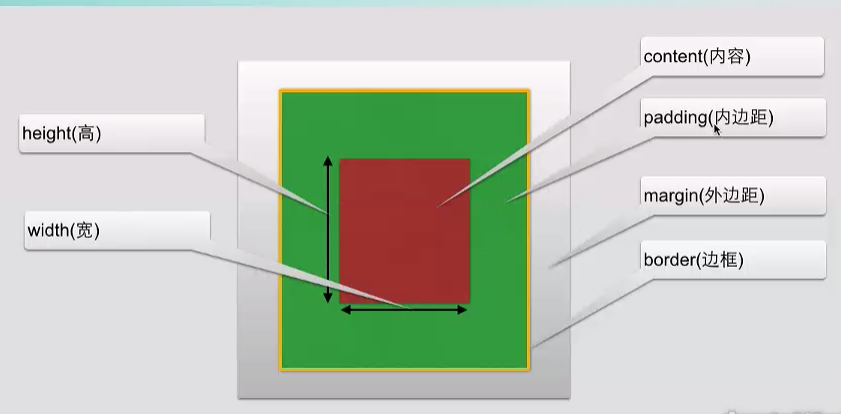
## 表格边框，文本对齐方式，背景色，宽高，边框重叠C:\Users\MIngdeng\Pictures\web课程截图\表格边框，文本对齐方式，背景色，宽高，边框重叠.PNG

## Css轮廓

## C:\Users\MIngdeng\Pictures\web课程截图\捕获6.PNG

# Css盒子模型

内容范围，margin（外边距）、border（边框）、padding（内边距）、content（内容）



## Border样式

Border-style:

## 单边框样式

Border-top-style:单边框样式

## 边框宽度

Border-width

## 单边框宽度

Border-top-width

## 边框颜色

Border-color

## 单边框颜色

Border-top-color

## 圆角边框

Border-radius：像素

## 边框阴影

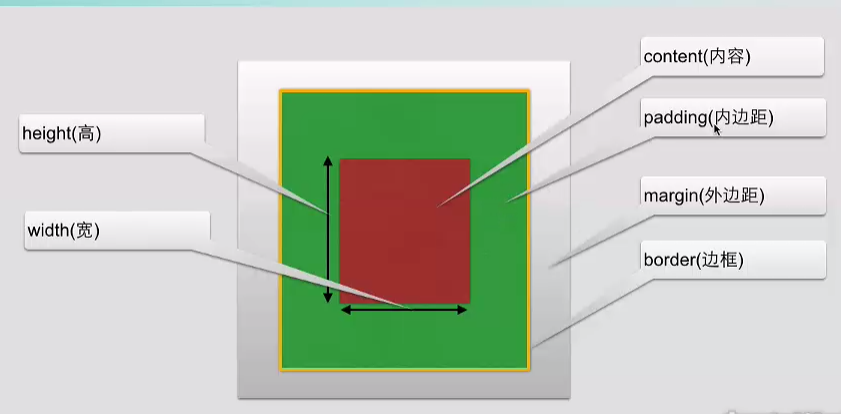
Box-shadow：向右偏移 向下偏移 透明度 颜色；

## 边框图片

Border-image

# Css外边距合并margin

两个盒子外边距相叠时可覆盖



# CSS定位

## Css position属性

Static静态布局

Relative相对布局

Absolute绝对布局

Fixed固定布局

## Css z-index属性

更改重叠呈现顺序,z-index:1;(适用于绝对布局和相对布局)

# Css选择器

## 元素选择器

根据元素标配使用< p style=”color:red”></p>

## 选择器分组

<h1><h2>

H1,h2{ \*\*\*\*\*}

## 使用通配符\*

## \*{ color:red}

## 类选择器

.class

结合元素选择器a.class

多类选择器.class1.class2

<p class=”p1 p2”>Hello</p>

Css:

.p1.p2{ color:red}

## Id选择器

## 简单属性选择器 [href]{}; [title]{}

通过值确定

1<p title=””>Hello</p>

2<p title=”tit”>Hello</p>

3<p title=”title”>Hello</p>

4<p title=”title hello”>Hello</p>

<style>

[title]{ color:red }/\*只有1应用此效果\*/

[titlt=”tit”]{ color:red }/\*只有2应用此效果\*/

[title~=”title”]{ color:red }/\*3,4都应用此效果,~=代表模糊等于\*/

## 后代选择器

<p>hello<strong>TOM</strong></p>

<style> p strong{…;}</style>通过空格隔开,可以隔代应用

## 子元素选择器

<p>hello<strong>TOM</strong></p>

<style> p>strong{…;}</style> 通过>传递,不可以隔代应用

## 相邻兄弟选择器

可选择金姐在另一个元素后的元素,且围着有相同父元素

例如: h1+p{}

# Css常用操作

## 对齐操作

### Margin水平对齐

### Position左右对齐

<style> div{ position:absolute; left:0px }/\*靠左对齐\*/</style >

### Float对齐

# Css动画

## 2d \3d转换

## 

## 通过transform引入

## translate移动 rotate旋转 scale缩放 skew倾斜

## transform:translate(100px;100px);x轴偏移量,y轴偏移量

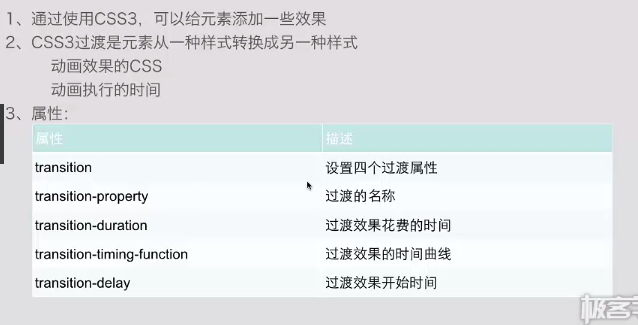
## transform:rotate(200deg);旋转200度

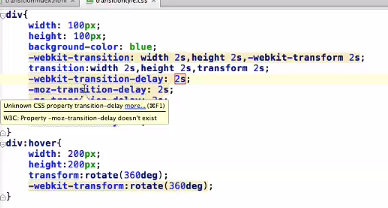
## transform:scale(1,2);缩放宽的倍数,缩放高的倍数

## transform:skew(50deg,50deg);x轴旋转度数,y轴旋转度数

## 

## Css过渡





Div{width:100x; height:100px; transition: width 2s, height 2s, transform 2s;}

/\*transition设置4个过度属性:wid宽度过度完成花费时间2s,hei,tra动画过度完成花费时间2s\*/

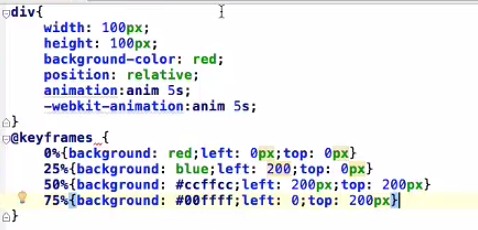
Div:hover{ width:200px; height:200px;transform:rotate(360deg)}

/\*m移入过度效果:…, …, 动画旋转360度\*/

## 动画

Css3的动画需要遵循@keyframes规则:

规定动画的时长 规定动画名称



## 定义和用法

通过 @keyframes 规则，您能够创建动画。

创建动画的原理是，将一套 CSS 样式逐渐变化为另一套样式。

在动画过程中，您能够多次改变这套 CSS 样式。

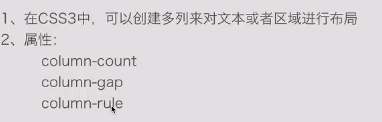
以百分比来规定改变发生的时间，或者通过关键词 "from" 和 "to"，等价于 0% 和 100%。

0% 是动画的开始时间，100% 动画的结束时间。

为了获得最佳的浏览器支持，您应该始终定义 0% 和 100% 选择器。

**注释：**请使用动画属性来控制动画的外观，同时将动画与选择器绑定。

## 多列



column-count 规定元素应该被分隔的列数。

column-gap 规定列之间的间隔

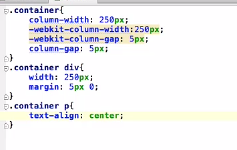
column-rule 设置所有 column-rule-\* 属性的简写属性。 3

column-rule-color 规定列之间规则的颜色。 3

column-rule-style 规定列之间规则的样式。 3

column-rule-width 规定列之间规则的宽度。

## Css瀑布流效果

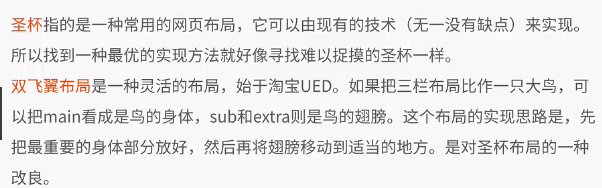


# 面向对象的css(OO CSS)

# Less的介绍和应用(css预处理器)

# **SASS 介绍和应用**

# 双飞翼布局(div+css布局)



CSS3基础

# **CSS3 文字与字体相关样式**

## Css3给文字添加阴影

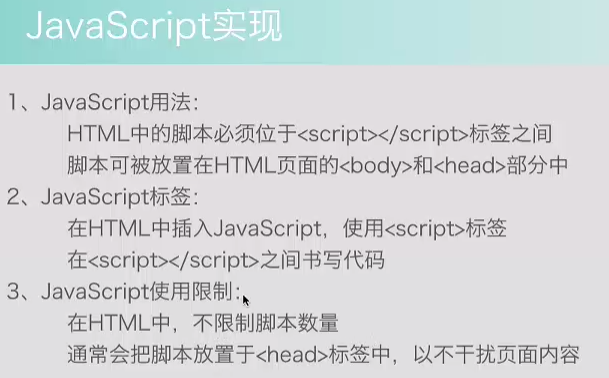
Text-shadow: length length length color;

可以添加多重阴影

<!—阴影离开文字的横向距离 阴影离开文字的纵向距离 模糊程度 阴影颜色 -->

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>

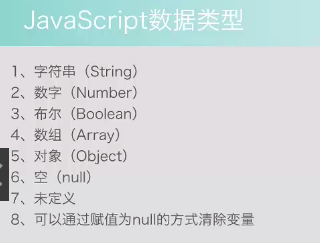
## 语法和注释

执行顺序按照顺序执行编写

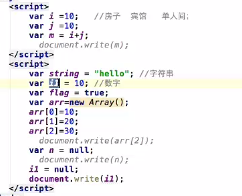
单行注释: //

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);

}

}

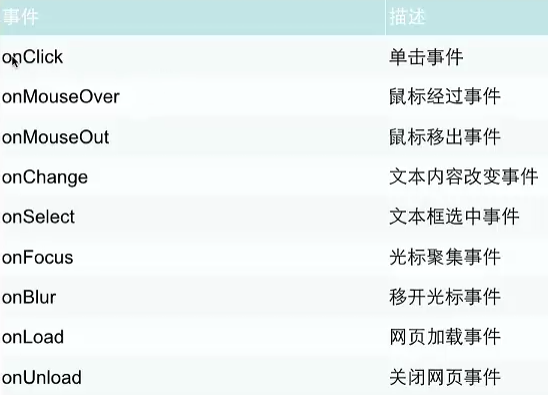
demo();

</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");

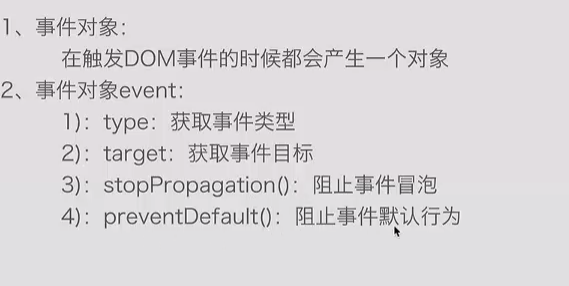
parent.removeChild(child);

</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

# jQuery教程

## jQuery 库 - 特性

jQuery 是一个 JavaScript 函数库。

jQuery 库包含以下特性：

* HTML 元素选取
* HTML 元素操作
* CSS 操作
* HTML 事件函数
* JavaScript 特效和动画
* HTML DOM 遍历和修改
* AJAX
* Utilities

## 向您的页面添加 jQuery 库

jQuery 库位于一个 JavaScript 文件中，其中包含了所有的 jQuery 函数。

可以通过下面的标记把 jQuery 添加到网页中：

<head>

<script type="text/javascript" src="jquery.js"></script>

</head>

请注意，<script> 标签应该位于页面的 <head> 部分。

## jQuery下载

下载地址：<http://jquery.com/download/>

## 通过 CDN（内容分发网络） 引用它。

引用Google CDN

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.0/jquery.min.js">

</script>

</head>

引用Microsoft CDN

<head>

<script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.8.0.js">

</script>

</head>

## jQuery语法

通过 jQuery，您可以选取（查询，query） HTML 元素，并对它们执行“操作”（actions）

jQuery 语法是为 HTML 元素的选取编制的，可以对元素执行某些操作。

基础语法是：$(selector).action()

美元符号定义 jQuery

选择符（selector）“查询”和“查找” HTML 元素

jQuery 的 action() 执行对元素的操作

示例

### $(this).hide() - 隐藏当前元素

### $("p").hide() - 隐藏所有段落

### $(".test").hide() - 隐藏所有 class="test" 的所有元素

### $("#test").hide() - 隐藏所有 id="test" 的元素

### 文档就绪函数$(document).ready(function(){…})

这是为了防止文档在完全加载（就绪）之前运行 jQuery 代码

## jQuery选择器

### jQuery元素选择器

jQuery 使用 CSS 选择器来选取 HTML 元素。

$("p") 选取 <p> 元素。

$("p.intro") 选取所有 class="intro" 的 <p> 元素。

$("p#demo") 选取所有 id="demo" 的 <p> 元素。

### jQuery属性选择器

jQuery 使用 XPath 表达式来选择带有给定属性的元素。

$("[href]") 选取所有带有 href 属性的元素。

$("[href='#']") 选取所有带有 href 值等于 "#" 的元素。

$("[href!='#']") 选取所有带有 href 值不等于 "#" 的元素。

$("[href$='.jpg']") 选取所有 href 值以 ".jpg" 结尾的元素。

### jQuery CSS 选择器

下面的例子把所有 p 元素的背景颜色更改为红色：

$(“p”).css(“background-color”,”red”);;;;;;;;

### more

$("ul li:first") 每个 <ul> 的第一个 <li> 元素

$("[href$='.jpg']") 所有带有以 ".jpg" 结尾的属性值的 href 属性

$("div#intro .head") id="intro" 的 <div> 元素中的所有 class="head" 的元素

$(".intro.demo") 所有 class="intro" 且 class="demo" 的元素

:first $("p:first") 第一个 <p> 元素

:last $("p:last") 最后一个 <p> 元素

:even $("tr:even") 所有偶数 <tr> 元素

:odd $("tr:odd") 所有奇数 <tr> 元素

http://www.w3school.com.cn/jquery/jquery\_ref\_selectors.asp

## jQuery事件

### jQuery名称冲突

jQuery 使用 $ 符号作为 jQuery 的简介方式。

某些其他 JavaScript 库中的函数（比如 Prototype）同样使用 $ 符号。

jQuery 使用名为 noConflict() 的方法来解决该问题。

var jq=jQuery.noConflict()，帮助您使用自己的名称（比如 jq）来代替 $ 符号。

亲自试一试

## jQuery效果

### jQuery toggle()(显示和隐藏) hide() 和 show()

//jQuery hide() & show()

$(document).ready(function(){

$("#show").click(function(){

$("#p1").show("slow",function(){$("#p1").hide("fast");});//$(selector).show(speed,callback); speed规定显示速度（参数有slow.慢.，fast.快. ，或者毫秒），callback规定显示完成后所执行的函数

});

$("#hide").click(function(){

$("#p1").hide("fast");

});

$("#toggle").click(function(){

$("#p1").toggle(1000);//使用 toggle() 方法来切换 hide() 和 show() 方法

});

});

### jQuery Fading()(淡入淡出)方法,

$("#fadeIn").click(function(){

$("#red").fadeIn("slow");//控制淡入已经隐藏元素，参数与toggle一样（slow，fast）

$("#blue").fadeIn("fast");

$("#yello").fadeIn(2000);

});

$("#fadeOut").click(function(){

$("#red").fadeOut("slow");

$("#blue").fadeOut("fast");

$("#yello").fadeOut(2000);

});

$("#fadeToggle").click(function(){

$("#red").fadeToggle("slow");//淡入或者淡出

$("#blue").fadeToggle("fast");

$("#yello").fadeToggle(2000);

});

$("#fadeTo").click(function(){

$("red").fadeTo("slow",0.15);//fadeTo给定渐变透明（参数：速度，透明度）

$("#blue").fadeTo("fast",0.6);

$("#yello").fadeTo(2000,0.9);

});

### jQuery 滑动方法

slideDown()//向下滑动

slideUp()//向上滑动

slideToggle()向上/向下切换

### jQuery animate()动画效果)

语法：

$(selector).animate({params},speed,callback);

必需的 params 参数定义形成动画的 CSS 属性。

可选的 speed 参数规定效果的时长。它可以取以下值："slow"、"fast" 或毫秒。

可选的 callback 参数是动画完成后所执行的函数名称。

下面的例子演示 animate() 方法的简单应用；它把 <div> 元素移动到左边，直到 left 属性等于 250 像素为止：

默认地，所有 HTML 元素都有一个静态位置，且无法移动。

如需对位置进行操作，要记得首先把元素的 CSS position 属性设置为 relative、fixed 或 absolute！

jQuery animate() - 使用相对值

也可以定义相对值（该值相对于元素的当前值）。需要在值的前面加上 += 或 -=：

//jQuery animate()动画

$("#animate").click(function(){

$("#animateDiv").animate({//animate属性，语法：$("selector").animate({params},speed,callback);

width:'800px',

left:'1200px'

},"slow",function(){

$("#animateDiv").animate({

width:'400px',

left:'0px'

},"slow");

});

});

$("#animate2").click(function(){//队列执行动画,按照顺序执行

$("#red,#blue,#yello").animate({

height:'800px',

opacity:'0.5'

},"slow");

$("#red,#blue,#yello").animate({

width:'800px',

opacity:'0.8'

},"slow");

$("#red,#blue,#yello").animate({

height:'400px',

opacity:'0.5'

},"slow");

$("#red,#blue,#yello").animate({

width:'400px',

opacity:'1'

},"slow");

});

### jQuery stop() 方法

jQuery stop() 方法用于停止动画或效果，在它们完成之前。

stop() 方法适用于所有 jQuery 效果函数，包括滑动、淡入淡出和自定义动画。

语法

$(selector).stop(stopAll,goToEnd);

可选的 stopAll 参数规定是否应该清除动画队列。默认是 false，即仅停止活动的动画，允许任何排入队列的动画向后执行。

可选的 goToEnd 参数规定是否立即完成当前动画。默认是 false。

因此，默认地，stop() 会清除在被选元素上指定的当前动画。

### jQuery Callback 函数

当动画 100% 完成后，即调用 Callback 函数。

典型的语法：

$(selector).hide(speed,callback)

*callback* 参数是一个在 hide 操作完成后被执行的函数。

### jQuery-chaning()

下面的例子把 css(), slideUp(), and slideDown() 链接在一起。"p1" 元素首先会变为红色，然后向上滑动，然后向下滑动：

$("#p1").css("color","red").slideUp(2000).slideDown(2000);

## jQuery 效果函数

|  |  |
| --- | --- |
| **方法** | **描述** |
| [animate()](http://www.w3school.com.cn/jquery/effect_animate.asp) | 对被选元素应用“自定义”的动画 |
| [clearQueue()](http://www.w3school.com.cn/jquery/effect_clearqueue.asp) | 对被选元素移除所有排队的函数（仍未运行的） |
| delay() | 对被选元素的所有排队函数（仍未运行）设置延迟 |
| dequeue() | 运行被选元素的下一个排队函数 |
| [fadeIn()](http://www.w3school.com.cn/jquery/effect_fadein.asp) | 逐渐改变被选元素的不透明度，从隐藏到可见 |
| [fadeOut()](http://www.w3school.com.cn/jquery/effect_fadeout.asp) | 逐渐改变被选元素的不透明度，从可见到隐藏 |
| [fadeTo()](http://www.w3school.com.cn/jquery/effect_fadeto.asp) | 把被选元素逐渐改变至给定的不透明度 |
| [hide()](http://www.w3school.com.cn/jquery/effect_hide.asp) | 隐藏被选的元素 |
| queue() | 显示被选元素的排队函数 |
| [show()](http://www.w3school.com.cn/jquery/effect_show.asp) | 显示被选的元素 |
| [slideDown()](http://www.w3school.com.cn/jquery/effect_slidedown.asp) | 通过调整高度来滑动显示被选元素 |
| [slideToggle()](http://www.w3school.com.cn/jquery/effect_slidetoggle.asp) | 对被选元素进行滑动隐藏和滑动显示的切换 |
| [slideUp()](http://www.w3school.com.cn/jquery/effect_slideup.asp) | 通过调整高度来滑动隐藏被选元素 |
| [stop()](http://www.w3school.com.cn/jquery/effect_stop.asp) | 停止在被选元素上运行动画 |
| [toggle()](http://www.w3school.com.cn/jquery/effect_toggle.asp) | 对被选元素进行隐藏和显示的切换 |

## jQuery HTML

### jQuery获取

#### 获得内容 - text()、html() 以及 val()

三个简单实用的用于 DOM 操作的 jQuery 方法：

text() - 返回所选元素的文本内容$("#btn1").click(function(){

alert("Text: " + $("#test").text());

});

* html() - 设置或返回所选元素的内容（包括 HTML 标记）
* val() - 设置或返回表单字段的值

#### 获取属性 - attr()

jQuery attr() 方法用于获取属性值。

下面的例子演示如何获得链接中 href 属性的值：$(“selector”).attr(“href”);

### jQuery 设置内容和属性

三个简单实用的用于 DOM 操作的 jQuery 方法：

text() - 设置所选元素的文本内容$("#btn1").click(function(){

$(“text”).text(“hello world”);

});

* html() - 设置或返回所选元素的内容（包括 HTML 标记）
* val() - 设置或返回表单字段的值

jQuery attr() 方法用于获取属性值。

下面的例子演示如何获得链接中 href 属性的值：$(“selector”).attr(“href”,”修改后的值”);

### text()、html() 以及 val() 的回调函数

上面的三个 jQuery 方法：text()、html() 以及 val()，attr()同样拥有回调函数。回调函数由两个参数：被选元素列表中当前元素的下标(i)，以及原始（旧的）值(origText)。然后以函数新值返回您希望使用的字符串。

$("#btn2").click(function(){

$("#test2").html(function(i,origText){

return "Old html: " + origText + " New html: Hello <b>world!</b>

(index: " + i + ")";

});

});

### jQuery添加元素

添加新的 HTML 内容

我们将学习用于添加新内容的四个 jQuery 方法：

* append() - 在被选元素的结尾nei插入内容$(“ol”).append(“<li>hello</ol>”);
* prepend() - 在被选元素的开头插入内容
* after() - 在被选元素之后插入内容
* before() - 在被选元素之前插入内容

### 添加若干新元素

function appendText()

{

var txt1="<p>Text.</p>"; // 以 HTML 创建新元素

var txt2=$("<p></p>").text("Text."); // 以 jQuery 创建新元素

var txt3=document.createElement("p"); // 以 DOM 创建新元素

txt3.innerHTML="Text.";

$("p").append(txt1,txt2,txt3); // 追加新元素

}

### jQuery删除元素

删除元素/内容

如需删除元素和内容，一般可使用以下两个 jQuery 方法：

* remove() - 删除被选元素（及其子元素）

下面的例子删除 class="italic" 的所有 <p> 元素：

实例

$("p").remove(".italic");

* empty() - 从被选元素中删除子元素

### jQuery 操作 CSS

jQuery 拥有若干进行 CSS 操作的方法。我们将学习下面这些：

* addClass() - 向被选元素添加一个或多个类($(“#p1”).addClass(“class1 class2”);)
* removeClass() - 从被选元素删除一个或多个类
* toggleClass() - 对被选元素进行添加/删除类的切换操作
* css() - 设置或返回样式属性

### jQuery CSS() 方法

获取css属性值

$(“#p1”).css(“background-color”);

设置css属性

$(“#p1”).css(“background-color”,”#ff0000”);

设置多个css属性

$(“p1”).css({“background-color”:”red”,”font-size”:”12px”});

### jQuery 尺寸 方法

jQuery 提供多个处理尺寸的重要方法：

* width()//可以设置可以返回值,不包含padding,border,margin
* height()//可以设置可以返回值,不包含p,b,m
* innerWidth()//不可以设置可以返回值,不包含p,b,m
* innerHeight()//不可以设置可以返回值,不包含p,b,m
* outerWidth()//不可以设置可以返回值,不包含p,b,m
* outerHeight()//不可以设置可以返回值,不包含p,b,m