

Javascript基础

脚本编写



2016-11-11

JavaScript笔记

貴社ますますご盛栄のこととお慶び申し上げます。平素は格別のご高配を賜り、厚く御礼申し上げます。

### 目录页

目录

[目录页 1](#_Toc467830470)

[1 语法 2](#_Toc467830471)

[1.1空的链接 2](#_Toc467830472)

[1.2 引用js脚本 2](#_Toc467830473)

[1.3 变量及数据类型 2](#_Toc467830474)

[1.3.1 类型转换 2](#_Toc467830475)

[1.3.2 字符串操作 2](#_Toc467830476)

[1.3.3 正则表达式 3](#_Toc467830477)

[1.3.4 正则表达式量词 3](#_Toc467830478)

[1.3.5 数组 3](#_Toc467830479)

[1.4 流程控制 ---for in 4](#_Toc467830480)

[1.5 函数 4](#_Toc467830481)

[1.6 对象 4](#_Toc467830482)

[2 浏览器对象模型 BOM 6](#_Toc467830483)

[2.1 对象类型 6](#_Toc467830484)

[Window对象：(定时器) 6](#_Toc467830485)

[History对象 6](#_Toc467830486)

[location 对象 6](#_Toc467830487)

[screen对象 6](#_Toc467830488)

[navigator 对象 7](#_Toc467830489)

[3 文档对象Dom 7](#_Toc467830490)

[3.1 列表框，下拉菜单操作： 8](#_Toc467830491)

[3.2 访问表格 8](#_Toc467830492)

[3.3 DOM对元素的增删改（IE不支持） 9](#_Toc467830493)

[3.4 New列表框、下拉菜单 9](#_Toc467830494)

[3.5 动态修改表格 9](#_Toc467830495)

[4 事件 10](#_Toc467830496)

[4.1 添加事件 10](#_Toc467830497)

[4.2 监听事件 10](#_Toc467830498)

[4.3 常用的事件 11](#_Toc467830499)

# 1 语法

## 1.1空的链接

<script>

<a href=”JavaScript:alert(“hello kitty”)”>点击</a>

</script> innerHTML= “小星星”;

<**video src="1.mp4" autoplay="autoplay" loop="loop"**></**video**>

## 1.2 引用js脚本

1. <script ***src*** = "move.js" type="text/javascript"></script>
2. <link rel="stylesheet" type="text/css" href="11-25.css"/>
3. <a href="javascript:go()">click here</a>

**document**.getElementById**("a")**.**innerHTML** = **"name:"**+stu.**name**+**"\nage:"**+stu.**age**;

**text-decoration**:**none**;

border: 1px solid gray;

background:url(<http://wyz.67ge.com/wp.jpg>);

**text-align**: **center**; **float**:**left**;

**display**: **none**; visibility:hidden

## 1.3 变量及数据类型

\* 隐式定义(使用前要赋值)

\*显示定义 Var temp； \_ $ 数字 字母

var i,j=”hello”;

* 换行

var temp = “hello \n kitty”;

var temp = “hello\n”+

“kitty”; NaN == not a number

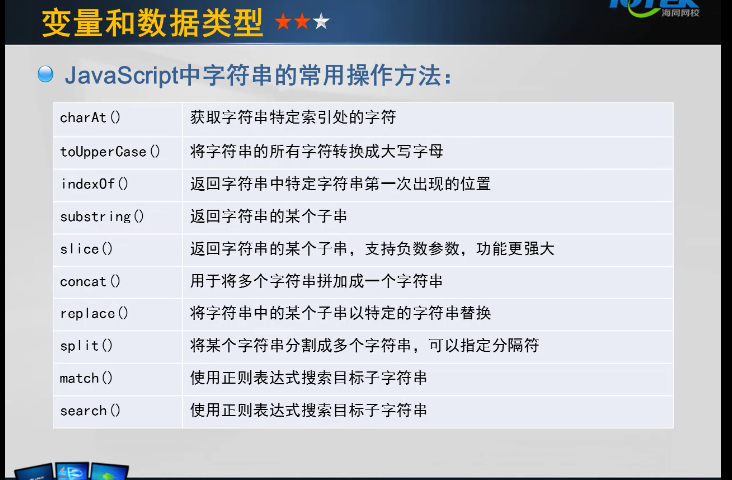
### 1.3.1 类型转换

var a = 3; var c = a.toString(); var c += a;

parseInt(“67red”) ==67; parseInt(“0xA”)//16进制可以转换

parseFloat(“35chen”) == 35.00 **无16进制转换**

### 1.3.2 字符串操作



返回下标索引

### 1.3.3 正则表达式

对字符串操作的逻辑公式，组成‘ **规则字符串**’；

特点：**灵活，逻辑性，功能性强，复杂；**

1. var reg = /chen/;
2. var reg = new RegExp(‘chen’);

exec(str) 返回找到的值；

reg.exec(str);

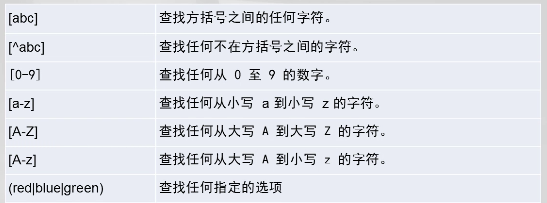
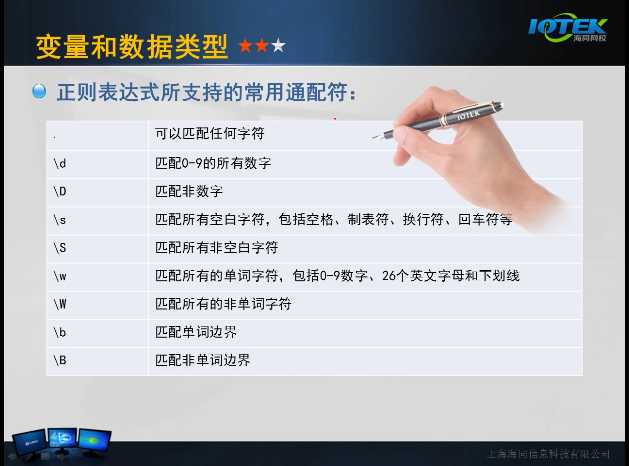
test(str) 返回 true；

通配符：

. 任意;

\d数字;\s空白字符;\w单词下划线;

o\b单词边界；（**判断有无，不在乎位置**）



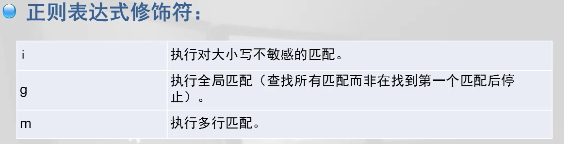
### 1.3.4 正则表达式量词



/[0-9]+/ i

邮箱表达式：

var reg = /^[0-9a-zA-Z\_-]+@[0-9a-zA-Z\_]+\.[com|cn|org]$/;



### 1.3.5 数组

var arr = [2,5,6];

var arr = [ ];

var arr = new Array(); **document.write(arr [i]+”<br/>”);**

**“10”==10; “10” !== 10;**

var a = **void**(b=0,c=3) ; undefine

string == **typeof** a; a **instanceof** Object == 1;

## 1.4 流程控制 ---for in

1 for (var index in arry){ } //输出的是赋值的数组下标。

2 for(var properties in navigator){ document.write(properties+":"+navigator[properties]+"<br/>");

}

throw new **Error**(“error”);

try{ if (1) { throw new **Error**(“error”); }}

catch(e){ document.write(e.message);}

finally{document.write(‘error’)};

## 1.5 函数

1 function show(){ } //可以先执行，后调用

2 var f = function (name){ } f(“bowen”); //匿名函数，要先定义后执行；

**(function (name))(‘bowen’);**

3 var f = new Function**(‘name’,’age’**,”alert(name, age)**;”**) ;

**函数默认是window的属性。**

调用：

1 window.fun();

2 fun.**call**(window,name,age ) ;

3 fun.**apply(** window, [‘zhangsan’,30] );

function show (arr,func){

**func.call(window,arr);**

}

show([1,2,3],function(arr){

for(i in arr){ document.write(arr[i]); }

})

## 1.6 对象

new Date(); new Date(month dd,yyy hh:mm:ss);

new Date(yyy,mth,dd//,hh,mm.ss) new Date(ms);

getDate(); getDay(); getMonth(); 0-11

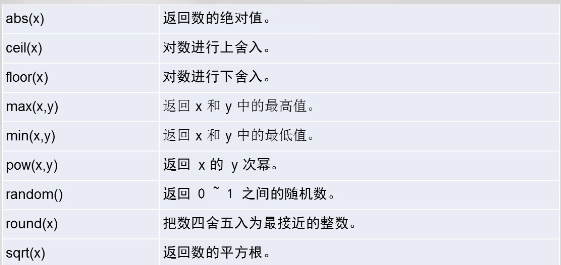
getFullYear(); getHours();0-23

getTime() 1970-1-1到现在的毫秒数

var d1 = new Date(); d1.toString();

var d2 = new Date(“2009-12-12 12:20:20”)

document .write(d2.toString());



### 对象创建：

new Object() JSON: {“name”:”zhangsan”,”age”:30}

new:一般通过构造函数添加属性；

Object：动态添加属性。（少）

Json：对象里，添加json对象数组

var Js = {

name:"zhangsan",age:30,

**son:[**

{name:"bowne",age:10},

{name:"nat",age:10}

**]**,

info:function(){document.write(this.name+":"+this.age+"<br/>");

for(var child in this.son){ document.write("son:"+this.son[child].name+","+this.son[child].age+"<br/>"); //document.write(child);}

}

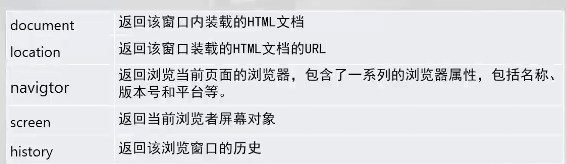
}

Js.info();

# 2 浏览器对象模型 BOM

## 2.1 对象类型

### Window对象：(定时器)

navigator

flag = confirm(“是否退出”)

flag=prompt(“密码”,”123”)

**moveBy(50,50)表示：+**

scrollBy(50,50)x,y

Tip:

**<pre>域格式化**

window.open([http://www.baidu.com,”baidu”,”left=20”);//](http://www.baidu.com,)多次打开，会重新加载

**location=no,toolbar=no,resizable=no; 地址栏 状态栏 大小（不推荐）**

var t = window.setInterval(“showTime()”,100); window调用方法：

window.clearInteral(t); **定时器，及清除定时器 窗口位置，滚动条，弹框，定时器**

### History对象

back();

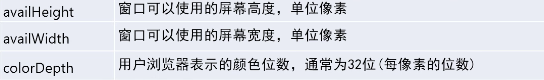
forward()//前进

go(intValue); go(-1) == back(); **history.back(); <br/> &nbsp**

### location 对象



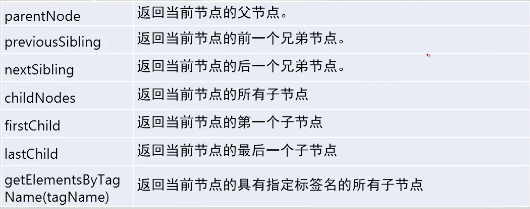
### screen对象



### navigator 对象



# 3 文档对象Dom



节点访问

form.submit()表单提交

form表单操作



表单元素查找3种

<form id="myform" action="http://www.baidu.com" method="get" target="\_blank">

<input name="username" type="text" value="nat"/><br/>

<input name="password" type="password" value="123456"/><br/>

<**select** name="city">

<option value="shanghai" selected="select">上海</option>

<option value="hefei">合肥</option>

</select><br/>

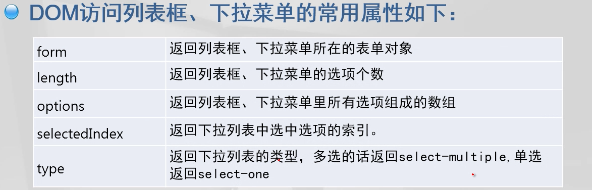
<input type="button" value="获取第一个表单" onclick="alert(document.getElementById('myform').elements[3].value)"

onclick="alert(document.getElementById('myform').elements[‘city’].value)"

onclick="alert(document.getElementById('myform').city.value)"/>

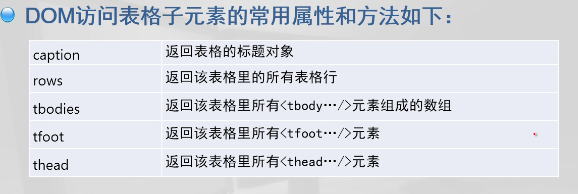
</form>

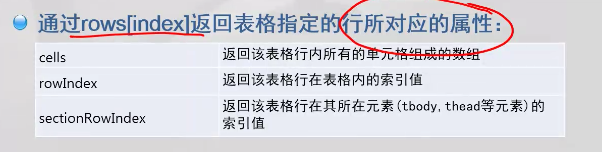
## 3.1 列表框，下拉菜单操作：



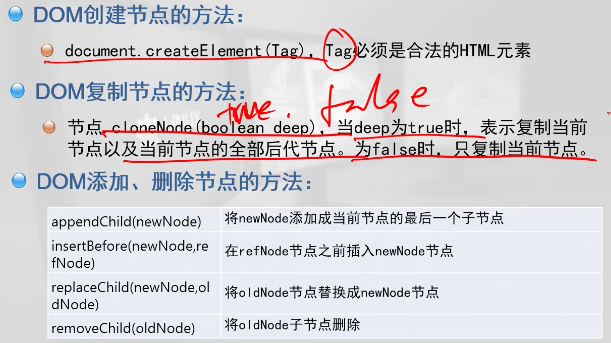


## 3.2 访问表格

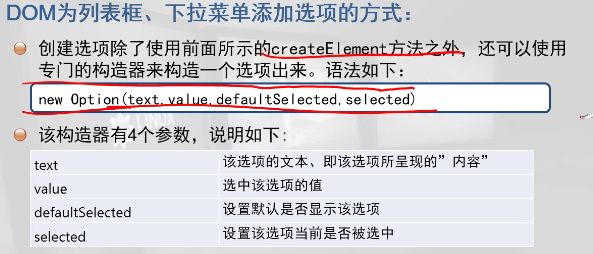




## 3.3 DOM对元素的增删改（IE不支持）



## 3.4 New列表框、下拉菜单



## 3.5 动态修改表格

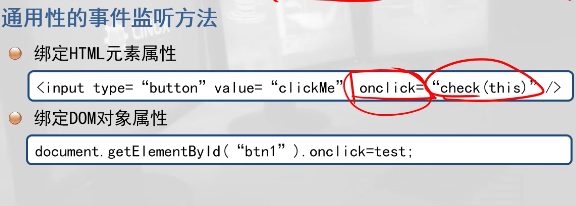




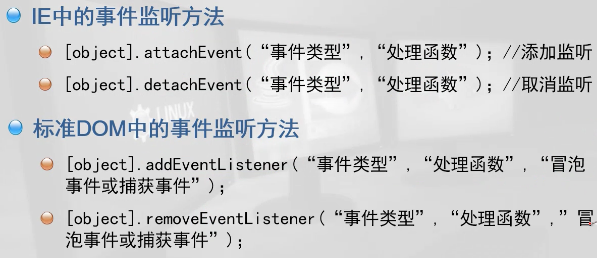
# 4 事件

事件：冒泡，捕获型事件（上到下，IE不支持）

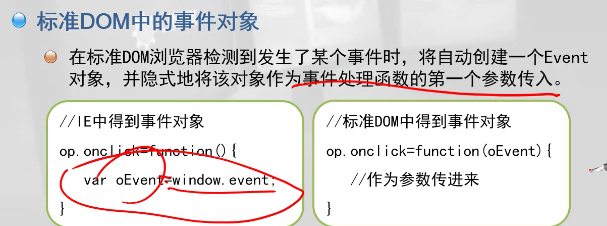
## 4.1 添加事件

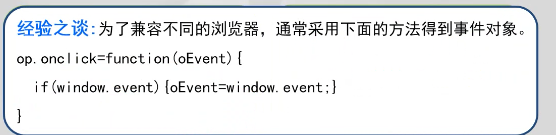


## 4.2 监听事件







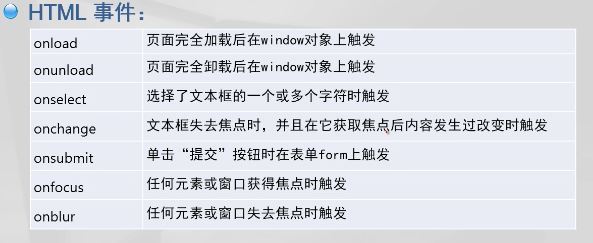




## 4.3 常用的事件







<http://www.xxdaohuaxiang.com/vod-type-id-1-pg-1.html>

# 5 实践之arcticle

h1:after

{

content:url(beep.wav);

}

onclick="window.scrollTo('0','0')"置顶

动画效果

width:100px;

height:100px;

background:yellow;

transition:height 2s;

**-moz-transition:width 2s; /\* Firefox 4 \*/**

**-webkit-transition:height 2s; /\* Safari and Chrome \*/**

-o-transition:height 2s; /\* Opera \*/

# 6 Jquery

## 下载使用

可以通过 CDN（内容分发网络） 引用它。

百度、又拍云、新浪、谷歌和微软的服务器都存有 jQuery 。

网站引用jQuery库：

|  |
| --- |
| <script src="http://cdn.static.runoob.com/libs/jquery/1.10.2/jquery.min.js"> |
| <script src="http://libs.baidu.com/jquery/1.10.2/jquery.min.js">//百度CDN |
| 又拍云 CDN: <script src="http://upcdn.b0.upaiyun.com/libs/jquery/jquery-2.0.2.min.js"> |
| <script src="http://lib.sinaapp.com/js/jquery/2.0.2/jquery-2.0.2.min.js">//新浪CDN |

## 2．jQuery基本语法

### ready(function(){})

**$(selector).action()//$jQuery提示符**

$(this).hide() $(“p.test”).hide()

$(document).**ready(function(){ });**

**$(function(){ });**

### 鼠标事件及 hover

$**(function()**{  
 $**("#head")**.**mouseenter(function()**{  
 $**("a")**.hide**()**;  
 }**)** $**("#head")**.**mouseleave(function()**{  
 $**("a")**.show**()**;  
 }**)** }**)**;

$("#p1**").hover**(function(){$("#p1").hide();},

function(){$("#p1").show();}

)

### 获得焦点

<input type="text" name="email">

$("input").**focus**(function(){ $(this).css("background-color","#cccccc"); });

$("input").blur(function(){ $(this).css("background-color","#ffffff"); });

### 隐藏、显示及切换

$("button").click(function(){ $("p").hide (1000); });

$("button").click(function(){ $("p").show(); });

$("button").click(function(){ $("p").toggle(); })；

$(*selector*).toggle(*speed,callback*);

### 淡入淡出（很关键）

* fadeIn()
* fadeOut()
* fadeToggle()
* fadeTo() //$("#div3").fadeTo("slow",0.7);透明度

### 滑动切换

* slideDown()
* slideUp()
* slideToggle()

### 创建动画animate（stop）

$(*selector*).animate({*params*}*,speed,****callback***); callback在动画执行完时调用

$("div").animate({left:'250px'});

1. position 属性设置为 relative, fixed, 或 absolute!

**2.Camel 标记法书写: 使用 paddingLeft 而不是 padding-left，使用 marginRight 而不是 margin-right，等等。**

3.动画值设置为 "show"、"hide" 或 "toggle"

4. 多个属性：div.animate({height:'100px',**opacity:'0.4'**},"slow");

### 链—执行多个动画

**$("button").click(function(){**

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

**});**

### jQuery的DOM—文档对象模型操作

* text() - 设置或返回所选元素的文本内容
* 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
* **attr（“href”）**

alert($("#runoob").attr("href"));

可修改：$("#test2").html**("<b>Hello world!</b>**");

* 1. text(callback(){})

$("#test1").text(function(i,origText){ return "旧文本: " + **origText** + " 新文本: Hello world! (index: " + i + ")"; });

* 1. $("#runoob").attr("href","http://www.runoob.com/jquery");

$("#runoob").attr({ "href" : "http://www.runoob.com/jquery", "title" : "jQuery 教程" });

$("#runoob").attr("href", function(i,origValue){ return origValue + "/jquery"; });

### DOM元素追加append()

* append() - 在被选元素的结尾插入内容
* 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

function appendText() { var txt1="<p>文本。</p>"; // 使用 HTML 标签创建文本 var txt2=$("<p></p>").text("文本。"); // 使用 jQuery 创建文本 var txt3=document.createElement("p"); txt3.innerHTML="文本。"; // 使用 DOM 创建文本 text with DOM

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

* addClass() - 向被选元素添加一个或多个类
* 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

*$("p").css({"background-color":"yellow","font-size":"200%"});*

* remove() - 删除被选元素（及其子元素）
* empty() - 从被选元素中删除子元素

### jQuery DimensionsjQuery尺寸

* width()
* height()
* innerWidth()
* innerHeight()
* outerWidth()
* outerHeight()

### 遍历

* parent()
* parents()
* parentsUntil()
* children()
* find()

### 遍历过滤

first()

last()

eq(0) –索引

$(“div p”).first();

* 1. **$("p").filter(".url");//匹配标准**
  2. $("p").not(".url");//匹配不标准

# 7 jQuery – Ajax（异步 Js xmL）

来个梨子

$("#div1").load("demo\_test.txt， **#p1**");

### 实现方法：load（）

$(selector).load(URL,data,callback);

data：过滤类，或是id

* *responseTxt* - 包含调用成功时的结果内容
* *statusTXT* - 包含调用的状态
* *xhr* - 包含 XMLHttpRequest 对象

$("#div1").load("demo\_test.txt",function(***responseTxt***,***statusTxt***,xhr){ if(statusTxt=="success") alert("外部内容加载成功!"); if(statusTxt=="error") alert("Error: "+xhr.status+": "+**xhr.statusText**); });

这是示例文件（"demo\_test.txt"）的内容：

<h2>jQuery AJAX 是个非常棒的功能！</h2> <p id="p1">这是段落的一些文本。</p>

### 两种方法

* *GET* - 从指定的资源请求数据

$.get(*URL*,*callback*);

大部分的函数都是有回调的

* *POST* - 向指定的资源提交要处理的数据

noConflict( ) 同以文档，使用多种模式

$.**noConflict();** //不使用$简写jQuery

jQuery(document).ready(function(){

jQuery("button").click(function(){ jQuery("p").text("jQuery 仍然在工作!"); });

});

或是，函数内部可以用$来简写jQuery

var jq = $.noConflict(); jq代替jQuery；

设计自己的符号代替jQuery

# Bootstrap

# 8 GIT指令操作

git config –list

git init

git status //git diff \*.cpp

1. git add .
2. git commit –m”new file”
3. git push

// git pull github.bowen...往本地下载

Git三步添加

* 1. git init
  2. git checkout --orphan gh-pages
  3. git add . git commit –m”xx”
  4. git remote add origin <https://github.com/username/jekyll_demo.git>
  5. **git push origin gh-pages**
  6. [**http://localhost:4000/**](http://localhost:4000/) **本地Hexo**

 hexo new "my new post"

* 1.  在H:\hexo\source\\_posts