目录

[一，js概述 1](#_Toc461022112)

[二、变量 2](#_Toc461022113)

[三、语句 3](#_Toc461022114)

[四、HTML DOM 5](#_Toc461022115)

[五、JS对象 8](#_Toc461022116)

## 一，js概述

1，js是面向对象的，解释型程序设计语言

最好不超过125k，js另外写减轻服务器压力

跨平台特性

2，js是运行在客户端

3，日常用途：

（1）嵌入在HTML页面，也可脱离

（2）对浏览器事件做出响应

（3）读写HTML元素

（4）在数据被提交到服务器之前验证数据

（5）检测访客的浏览器信息

（6）控制cookie

（7）基于Node.js技术进行服务器端编程

4，要求

div+css

周246晚20：30~22：30

2604168723

作业提交格式：0825-作业课题-yy昵称-有问题

844985059追梦邮箱

JavaScript是大小写敏感的。

## 二、变量

1，变量是储存信息的容器

2，变量可用于存放值和表达式

三、数据类型

1，number string array object boolean null undifined

对象：

var person={

firstname : "Bill",

lastname : "Gates",

id : 5566

};

对象属性有两种寻址方式：

name=person.lastname;

name=person["lastname"];

在javascript中，对象是数据（变量），拥有属性和方法。

javascript中，几乎所有的事务都是对象：字符串，数字，数组，日期，函数，等等。

toUpperCase( )字符串大写转换

2，函数是由事件驱动的或者当它被调用时执行的可重复利用的代码块。

局部变量在函数运行以后被删除；

全局变量在页面关闭后被删除。

## 三、语句

1，通过标签引用，break 语句可用于跳出任何 JavaScript 代码块

可以对 JavaScript 语句进行标记。

如需标记 JavaScript 语句，请在语句之前加上冒号：

label:

语句

<script>

cars=["BMW","Volvo","Saab","Ford"];

list:

{

document.write(cars[0] + "<br>");

break list;

document.write(cars[1] + "<br>");

document.write(cars[2] + "<br>");

document.write(cars[3] + "<br>");

}

</script>

2，try catch

try{

//运行代码

}

catch ( err ){

//处理错误

}

例子：

<script>

try{

alertt(“hhhhhhh”);

}

catch(err){

txt=”there was an error on this page .\n\n”;

txt+=”Error description:”+err.message+”\n\n”;

alert(“txt”);

}

</script>

3,throw语句

throw 语句允许我们创建自定义错误。

try{

var x=document.getElementById("input").value;

alert(x);

if(x=="1") throw "empty";

else if(isNaN(x)) throw "NaN";

}

catch(err){

alert("hhhh: "+err);

}

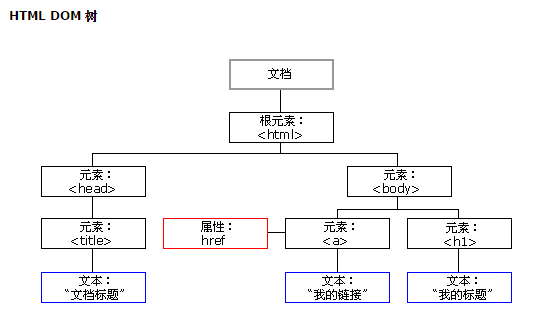
4，表单验证：

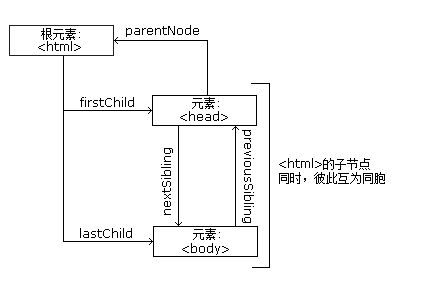
被 JavaScript 验证的这些典型的表单数据有：

* 用户是否已填写表单中的必填项目？
* 用户输入的邮件地址是否合法？
* 用户是否已输入合法的日期？
* 用户是否在数据域 (numeric field) 中输入了文本？

## 四、HTML DOM

DOM文档对象模型：





通过id名查找：

document.getElementById(“id”);

通过标签名查找：

document.getElementsByTagName(“tag”);

通过类名查找：//在IE5，6，7，8中无效

document.getElementsByClassName(“class”);

1，改变HTML内容：

document.getElementById(“id”).innerHTML=”txt”;

2，改变HTML属性：

document.getElementById(id).attribute=new value ;

document.getElementById(“id”).src=”1.jpg”;

3，改变HTML样式：

document.getElementById(“id”).style.color=”red”;

.style.visibility=”hidden”

style.visibility=”visible”

4，DOM事件

事件一般有：

当用户点击鼠标时；onclick

当页面已经加载时；document.ready(){ }

当图像已经加载时；

当鼠标移动到元素上时；

当输入字段被改变时；onchange

当提交HTML表单时；onsubmit

当用户触发按键时。

(1)onload 和 onunload 事件会在用户进入或离开页面时被触发。

(2)onchange 事件常结合对输入字段的验证来使用。

(3)onmouseover 和 onmouseout 事件可用于在用户的鼠标移至 HTML 元素上方或移出元素时触发函数。

(4)onmousedown, onmouseup 以及 onclick 构成了鼠标点击事件的所有部分。首先当点击鼠标按钮时，会触发 onmousedown 事件，当释放鼠标按钮时，会触发 onmouseup 事件，最后，当完成鼠标点击时，会触发 onclick 事件。

(5)onfocus onblur对象获得/失去焦点事件

5，节点操作

创建新的HTML元素：

document.createElement(“p”)

document.createTextNode(“xxxxx”)

appendChild()

<script>

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

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

para.appendChild(node);

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

element.appendChild(para);

</script>

删除已有的HTML元素：

如需删除 HTML 元素，您必须首先获得该元素的父元素

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

child.parentNode.removeChild(child);

## 五、JS对象

1，创建对象 { }

person=new Object();

person.firstname="Bill";

person.lastname="Gates";

person.age=56;

person.eyecolor="blue";

person={firstname:"John",lastname:"Doe",age:50,eyecolor:"blue"};

2，for…..in 语句循环遍历对象的属性

var person={name:"liqi",age:"21",sex:"female"};

for(var x in person){

alert(person[x]); //输出：liqi 21 female

alert(x); //输出：name age sex

}

3，字符串对象

使用 match() 来查找字符串中特定的字符，并且如果找到的话，则返回这个字符。

常用方法：

toUpperCase()

toLowerCase()

split() 把字符串分割为字符串数组。

indexOf() 检索字符串。

4，日期对象