[**js&dom&jQuery**](http://www.cnblogs.com/ciciah/articles/8395418.html)

JavaScript是一种属于网络的脚本语言,已经被广泛用于Web应用开发,常用来为网页添加各式各样的动态功能,为用户提供更流畅美观的浏览效果。通常JavaScript脚本是通过嵌入在HTML中来实现自身的功能的。

**1、存在方式**

[复制代码](javascript:void(0);)

复制代码

<!--导入javascript脚本方法-->

<script type="text/javascript" src="t1.js "></script>

<!--直接在html内部编写javascript-->

<script type="text/javascript">

function func() {

alert("Hello")

}

复制代码

[复制代码](javascript:void(0);)

**2、javascript代码块位置**

放在<body>标签内的代码底部，为什么不能放在上面呢？为什么css的就可以放在上面呢？

注：css代码首先要记住html代码是从上往下解释的，如果css代码放在下面，在上面的代码使用css样式，如果css代码块放在下面就没有办法显示样式了

另不同的浏览器处理请求也不同：正常来说当有一个请求过来时候会把js&css一起发送过去，咱们按照最low的方式理解的话可以这么理解：如果js文件或者js耗时比较久的话，下面的html代码就无法执行了。

**3、变量和函数的声明**

变量：

[复制代码](javascript:void(0);)

复制代码

function m1() {

alert("hello")

var name = 'yunyun'; //var 变量名 ,变量名前面加var为局部变量

age = '18';

//注这里需要注意,建议使用的时候一般不要使用全局变量!否则如果代码量比较大的

//时候容易出现调用混乱的问题

}

m1();

复制代码

[复制代码](javascript:void(0);)

函数

[复制代码](javascript:void(0);)

// 普通函数

function func() {

alert("Hello yunyun")

}

// 定义一个可传参数的函数

function func(arg) {

alert(arg)

[复制代码](javascript:void(0);)

[复制代码](javascript:void(0);)

**js 的展示方法有两种**：  
通过网页来展示

<script type="text/javascript">

function f1() {

alert("hello")

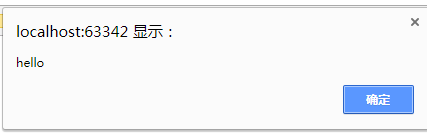
}

f1()

</script>

[复制代码](javascript:void(0);)

显示效果如下：



通过console来展示

<script type="text/javascript">

function f1() {

console.log("Hello")

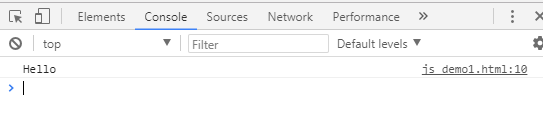
}

f1()

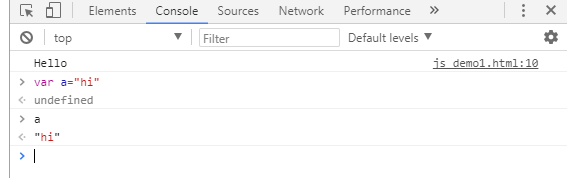
</script>

显示效果如下：

打开google chrome F12点击"Console"，刷新页面



调试的地方可以在google chrome 上进行测试，F12点击"Console"



**4、字符串常用方法及属性**

obj.trim()  去除空格

[复制代码](javascript:void(0);)

复制代码

var a = " hello "

undefined

a.trimLeft() #去除左边的空格

a.trimRight() #去除右边的空格

a.trim() //去除两边的空格

a #这里可以发现我执行了上面的去除空格的命令之后,实际的值是没有改变的

b = a.trim()

#但是我们可以通过赋值来改变他

b  
  
"hello"

复制代码

[复制代码](javascript:void(0);)

obj.charAt(index) 根据索引获取字符串里的字符

[复制代码](javascript:void(0);)

复制代码

b

"JSStudy"

b.charAt(0)

"J"

b.charAt(1)

"S"

b.charAt(2)

"S"

复制代码

[复制代码](javascript:void(0);)

obj.substring(start,end)  获取字符的子序列，类似于切片

b

"JSStudy"

b.substring(0,3)

"JSS"

obj.indexOf(char) 去字符串找指定的字符的索引值是多少

b

"hello"

b.indexOf("e")

1

obj.length  获取字符串的长度

b

"hello"

b.length

5

**5、数组**

声明一个数组和python中的列表类似

a = [11,22,33,44] #声明一个数组

[11, 22, 33, 44]

插入

[复制代码](javascript:void(0);)

复制代码

a = [11,22,33,44] #声明一个数组

[11, 22, 33, 44]

a.push(55) #在数组最后增加一个元素

5 #这里是数组的长度

a

[11, 22, 33, 44, 55]

a.unshift(00) #在数组最前面增加一个元素

6 #长度

a

[0, 11, 22, 33, 44, 55]

a.splice(3,0,'insert') #在指定的索引增加一个元素,括号内(3为索引值,0为固定值,要插入的内容)

[]

a

[0, 11, 22, "insert", 33, 44, 55]

a.unshift(100)

8

复制代码

[复制代码](javascript:void(0);)

移除

[复制代码](javascript:void(0);)

复制代码

a

[100, 0, 11, 22, "insert", 33, 44, 55]

a.pop() # 从尾部获取

55

a.shift() #从开头获取

100

a

[0, 11, 22, "insert", 33, 44]

a.splice(3,1) #从指定位置获取,括号内参数为(元素的索引,后面为索引后的元素个数,包含本身)

["insert"]

a

[0, 11, 22, 33, 44]

复制代码

[复制代码](javascript:void(0);)

切片

a

[0, 11, 22, 33, 44]

a.slice(1,3)

[11, 22]

a

[0, 11, 22, 33, 44]

合并

[复制代码](javascript:void(0);)

复制代码

a = [11,22]

[11, 22]

b = [44,55]

[44, 55]

a.concat(b)

[11, 22, 44, 55]

a

[11, 22]

b.concat(a)

[44, 55, 11, 22]

复制代码

[复制代码](javascript:void(0);)

反转

a

[11, 22]

a.reverse()

[22, 11]

a

[22, 11]

字符串格式化

a

[22, 11]

a.join('\_')

"22\_11"

a

[22, 11]

数组长度

a

[22, 11]

a.length

2

**6、字典**

字典是数组的一种特殊形式

dict1 = {'k1':123,'k2':234} #定义一个字典

Object {k1: 123, k2: 234}

dict1['k1']

123

**7、循环**

js中的循环有两种方式

[复制代码](javascript:void(0);)

复制代码

#第一种

for (var i=0;i<10;i++) {console.log(i)}

#输出结果,看本代码下第一图

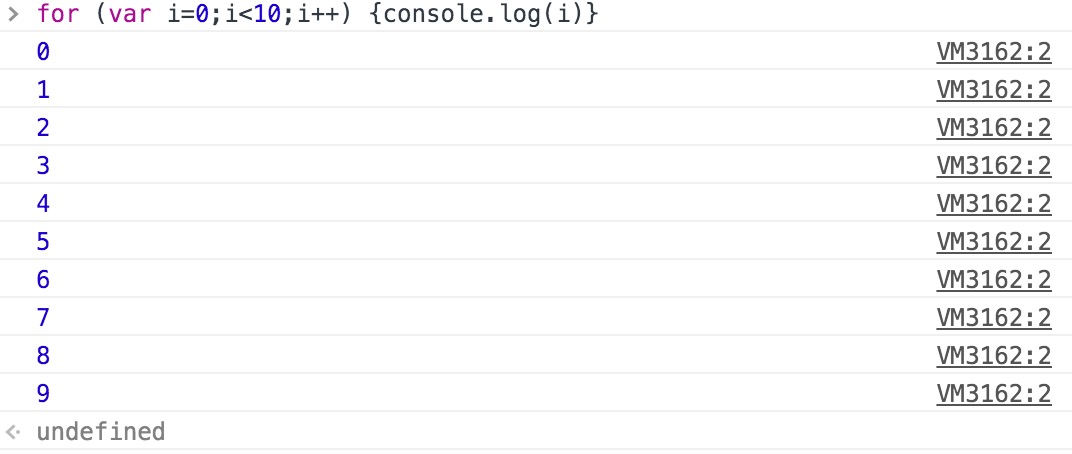
#第二种

for (var item in a) {console.log(a[item])}

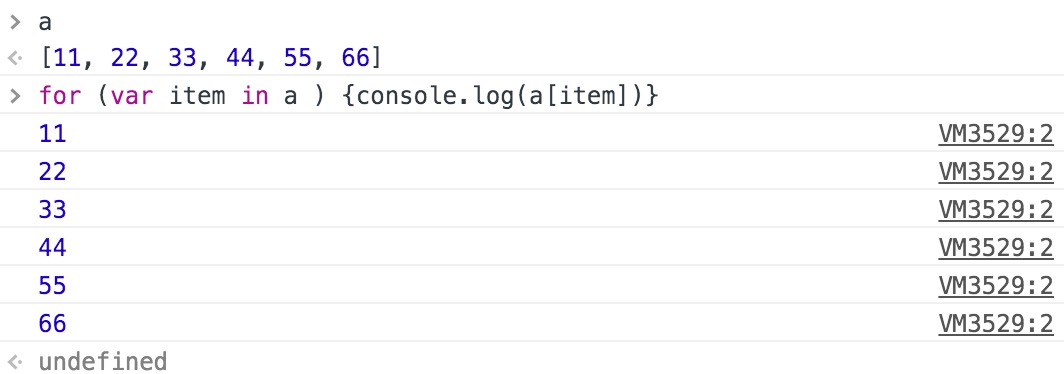
#输出结果，看本代码下第二图

复制代码

[复制代码](javascript:void(0);)



图二：



**8、异常处理**

和python中的异常处理类似，代码如下：

[复制代码](javascript:void(0);)

复制代码

<script type="text/javascript">

try{

var tiancai = isme

}catch(e) {

console.log(e)

}finally{

console.log("aaaa")

}

</script>

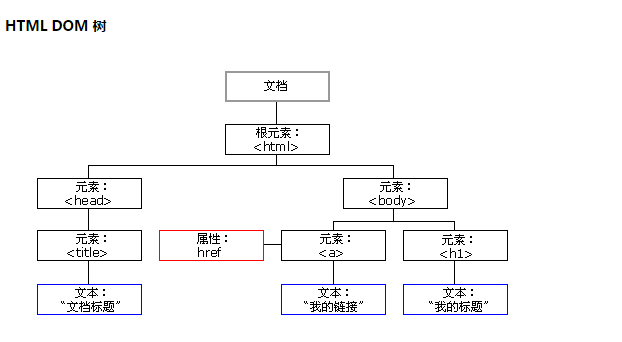
复制代码

[复制代码](javascript:void(0);)

**DOM编程**

文件对象模型（Document Object Model，简称DOM），是W3C组织推荐的处理可扩展标志语言的标准编程接口。

DOM编程：可以操作html所有的标签，进行找、操作！他也是javascript的一部分



**1、选择器：**

document.getElementById('id')  查找指定的id，然后我们可以进行操作

[复制代码](javascript:void(0);)

复制代码

<body>

<div id="id\_1">

123

</div>

<script type="text/javascript">

var i = document.getElementById('id\_1'); //查找指定的id

i.innerText = '456'; //innerText修改指定的字符串

</script>

</body>

复制代码

[复制代码](javascript:void(0);)

显示效果，当我们打开IE的时候123就会被修改为456

下面操作方式也类似：

document.getElementsByName('name')

[复制代码](javascript:void(0);)

复制代码

<body>

<div name="name\_1">

123

</div>

<script type="text/javascript">

var i = document.getElementsByName('name\_1'); //查找指定的name,这里和ID不一样name可以有多个

for (var item in i) {

i[item].innerText = '456'; //innerText修改指定的字符串

};

</script>

</body>

复制代码

[复制代码](javascript:void(0);)

document.getElementsByTagName('tagname')

[复制代码](javascript:void(0);)

复制代码

<body>

<div>

123

</div>

<div>

234

</div>

<script type="text/javascript">

var i = document.getElementsByTagName('div'); //查找指定的标签类型,这里同样和ID不一样标签比如<div>标签可以有多个

for (var item in i) {

i[item].innerText = '456'; //innerText修改指定的字符串

};

</script>

</body>

复制代码

[复制代码](javascript:void(0);)

上面的代码试用jquery会非常方便就不需要重造轮子

**2、注册 事件**

首先了解下面的意思：

事件：比如有一个“按钮”，当你点击的时候发生什么，双击的时候发生什么，这个就叫做事件！

注册：首先这个按钮，当你点击的时候发生的动作，上面的事件要使他出现我们想要的效果，首先得把事件和函数进行绑定，然后把他们注册到指定标签上。

常用事件：

* onclick
* onblur
* onfocus
* ..................

举例代码如下：

写一个input的标签，首先把事件和函数进行绑定，当你一点这个按钮的时候就会自动执行，绑定的函数：

[复制代码](javascript:void(0);)

复制代码

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>shuaige\_js\_file</title>

<!--导入javascript脚本方法-->

<!--<script type="text/javascript" src="t1.js "></script>-->

<style>

.color\_red {

background-color: red;

}

</style>

</head>

<body>

<div id="id\_1">

234

</div>

<!--下面的input标签:onclick是个事件他等于一个函数,就是事件和函数进行绑定,应用到标签中就是注册到标签中-->

<input type="button" onclick="edit();" value="修改" />

<script type="text/javascript">

function edit() {

var i = document.getElementById('id\_1');

i.className = 'color\_red';

}

</script>

</body>

</html>

复制代码

[复制代码](javascript:void(0);)

 那么恢复按钮呢？只要在新增一个即可

[复制代码](javascript:void(0);)

复制代码

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>shuaige\_js\_file</title>

<!--导入javascript脚本方法-->

<!--<script type="text/javascript" src="t1.js "></script>-->

<style>

.color\_red {

background-color: red;

}

</style>

</head>

<body>

<div id="id\_1">

234

</div>

<!--下面的input标签:onclick是个事件他等于一个函数,就是事件和函数进行绑定,应用到标签中就是注册到标签中-->

<input type="button" onclick="edit();" value="修改" />

<input type="button" onclick="rollback();" value="回滚"/>

<script type="text/javascript">

function edit() {

var i = document.getElementById('id\_1');

i.className = 'color\_red';

}

function rollback() {

var i = document.getElementById('id\_1');

i.className = ''; //这里只要给他设置为空即可

}

</script>

</body>

</html>

复制代码

[复制代码](javascript:void(0);)

 事件列表：

|  |  |
| --- | --- |
| 事件列表 | |
| 属性 | 此事件什么时候发生（什么时候被触发 |
| onabort | 图象的加载被中断 |
| onblur | 元素失去焦点 |
| onchange | 区域的内容被修改 |
| onclick | 当用户点击某个对象时调用的事件句柄（比点击input标签时执行上面的代码例子） |
| ondblclick | 当用户双击某个对象时调用的事件句柄 |
| onerror | 在加载文档或图像时发生错误 |
| onfocus | 元素获得焦点 |
| onkeydown | 某个键盘按键被按下 |
| onkeypress | 某个键盘按键被按下并松开 |
| onkeyup | 某个键盘被松开 |
| onload | 一张页面或一副图片完成加载 |
| onmousedown | 鼠标按钮被按下 |
| onmousemove | 鼠标移动过来后 |
| onmouseout | 鼠标从某个元素移开 |
| onmouseover | 鼠标移动到某个元素之上 |
| onmouseup | 鼠标按键被松开 |
| onreset | 重置按钮被点击 |
| onresize | 窗口或框架被重新调整大小 |
| onselect | 文本被选中 |
| onsubmit | 确认按钮被点击 |
| onunload | 用户退出页面 |

**注：一个标签可以绑定多个事件！！**

<input type="button" onmouseover="edit()" onmouseout="rollback()" value="修改&回滚" />

**注：onload 和其他的不太一样，他是写在Javascirpt里的**

[复制代码](javascript:void(0);)

复制代码

<script type="text/javascript">

// 这里的onload是,当整个页面加载完成之后才执行的,整个页面包含的元素加载完成之后才执行的.

window.onload = function () {

alert("The page Load complete")

};

function edit() {

var i = document.getElementById('id\_1');

i.className = 'color\_red';

}

function rollback() {

var i = document.getElementById('id\_1');

i.className = ''; //这里只要给他设置为空即可

}

</script>

复制代码

[复制代码](javascript:void(0);)

**jQuery**

jQuery是一个兼容多浏览器的javascript库，核心理念是write less,do more(写得更少,做得更多)，对javascript进行了封装，是的更加便捷的开发，并且在兼容性方面十分优秀。

要想使用jQuery首先得导入代码如下（附加简单应用）：

dom也很有用，对于了解jQuery有很大帮助

[复制代码](javascript:void(0);)

复制代码

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>shuaige</title>

<style>

.r {

background-color:red; /\* 给标签设置背景颜色为红色\*/

}

</style>

</head>

<body>

<div id="id\_1">

123

</div>

<div class="c1">1</div>

<div class="c1">2</div>

<div class="c1">3</div>

<!--导入Jquery文件-->

<script type="text/javascript" src="jquery-1.8.2.min.js"></script>

<script>

$(function () {alert('Hello shuaige');}); //当页面执行完后加载

//这里$是什么呢? 他可以理解为jQurey创建的对象,类似于python中的类创建的对象,对象去调用方法一样.!!!仅仅是类似

$('#id\_1').text('456');

//分解 $('#id\_1') 找到id为id\_1的标签,并把里面的内容修改为456

//这里虽然永不倒dom但是,会dom对jQurey有很大的帮助

$('.c1').addClass('r');

//分解 $('.c1') 找到class为c1的标签

</script>

</body>

</html>

复制代码

[复制代码](javascript:void(0);)

 就可以这么理解:**$('这部分是选择').操作(function () {} )**