

js输出
js语句与注释
JS数据类型
js变量
js函数
js运算符
js比较和逻辑运算符
js条件语句与switch语句
js中for循环与while循环
js中break和continue语句
DOM
DOM HTML
DOM CSS
DOM事件不同写法
DOM节点
JS window
window.screen
window Location
window.history

js输出

- `window.alert()`:警告框
- `document.write()`:写到HTML文档中
- `innerHTML`:写到HTML元素
- `console.log()`:写到浏览器的控制台

- ```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8" />
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>js输出</title>
7 </head>
8 <body>
9 <p id="one"></p>
10 <script type="text/javascript">
11
12 // window.alert():警告框
13 // document.write():写到HTML文档中
14 // innerHTML:写到HTML元素
15 // console.log():写到浏览器的控制台
16
17
18 // window.alert('hellowrold');
19
```

```

20 // document.write('hellowrold');
21
22 // document.getElementById('one').innerHTML="this is p";
23
24 console.log('helloworld');
25 </script>
26 </body>
27 </html>

```

## js语句与注释

- 分号
- JavaScript代码
- JavaScript语句标识符
- 代码块
- 单行和多行注释

## JS数据类型

- 字符串(String): `var string ="voasds sda";`
- 数字(Number): `var number =1;`
- 布尔(Boolean): `var x =true;`
- 数组(Array):

```

1 1.var cars=new Array();
2 cars[0]="savv";
3 2.var cars= new Array("saasd","dasd","dasd");
4 3.var cars=["sdasd","dasd","gdgd"];

```

- 对象(Object): `var person={firstname:"john",lastname:"doe",id:5566};`
- 空(null): 可以通过设置变量为空清空变量
- 未定义(Undefined): `var persons;` 这个值表示变量不含有值
- var可以声明7大类型

## js变量

- 变量必须以字母开头
- 变量也能以\$和下划线符号开头 (不推荐)
- 变量名称对大小写敏感

## js函数

- 函数是由事件驱动的或者当它被调用时执行的可重复使用的代码块
- `function a(参数){}` 声明会前置
- `var a =function(参数){}`
- 函数中的return

## js运算符

- 赋值运算符 `= += -=` 等
- 算术运算符 `+ - * /` 等
- 字符串的`+`运算符

## js比较和逻辑运算符

- `<` 小于
- `>` 大于
- `==` 等于1
- `===` 绝对等于（值和类型均相等）
- `!=` 不等于
- `&&` and
- `||` or

## js条件语句与switch语句

- 条件语句: `if else if else`
- switch语句: `switch case break default`

## js中for循环与while循环

- for - 循环代码块一定的次数
- for/in - 循环遍历对象
- while - 当指定的条件为true时循环指定的代码块

- ```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1">
6      <title></title>
7    </head>
8    <body>
9      <script type="text/javascript">
10
11        for(var i =0 ;i<10 ;i++){
12          console.log('i'+i);
13        }
14
15
16        var person ={
17          name: 'json',
18          age:25,
19          sg:170
20        }
```

```

21         for(key in person){
22             console.log(person[key])
23         }
24
25
26         var count =1
27         while(count <10){
28             console.log('执行');
29             count++;
30         }
31     </script>
32 </body>
33 </html>

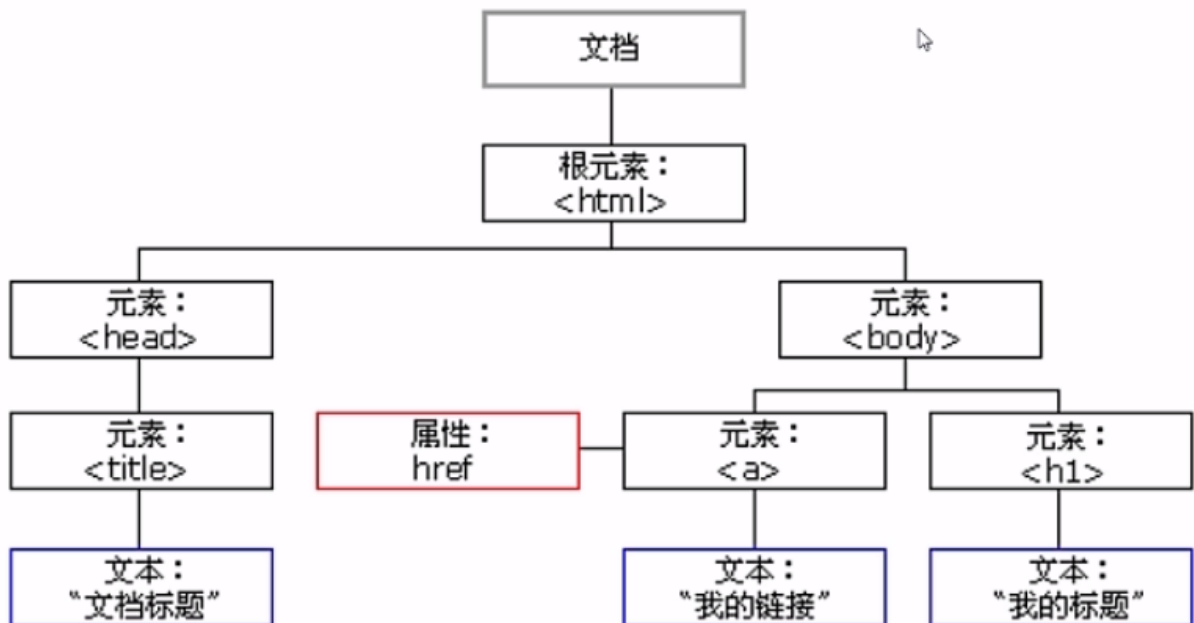
```

js中break和continue语句

- `break` 语句可用于跳出循环
- `continue` 语句跳出本次循环，继续执行

DOM

DOM树模型



- 获得元素
 - `document.getElementById()`
 - `document.getElementsByTagName()`
 - `document.getElementsByClassName()`

DOM HTML

- 修改HTML内容

- `document.getElementById(id).innerHTML = "helloWorld"`

- 修改HTML属性

- `element.getAttribute()`

- `element.setAttribute()`

- `element.src`

- `element.href`

- ```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8" />
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title></title>
7 </head>
8 <body>
9 <div id="main" data="nihao">123</div>
10
11
12 调到百度
13
14 <script type="text/javascript">
15 var main= document.getElementById("main");
16
17
18
19 main.innerHTML = "helloworld";
20 //获取data属性的值
21 alert(main.getAttribute("data"));
22
23 /**
24 * 1.判断是否有data值
25 * 2.有data值修改data值为buhao
26 * 3.如果没有这个属性则会添加这个属性
27 */
28 main.setAttribute("data","buhao");
29 main.setAttribute("dd","dasd");
30 document.write(main.getAttribute("data"));
31 document.write(main.getAttribute("dd"));
32
33
34 var img =document.getElementById("img");
35 img.src="2.jpg";
36
37 var goUrl=document.getElementById("goUrl");
38 goUrl.href="www.baidu.com"
39 </script>
40 </body>
41 </html>
```

## DOM CSS

- 修改样式

- `document.getElementById('id1').style.color='blue'`

## DOM事件不同写法

- `<h1 onclick="this.innerHTML='谢谢'">请点击该文本</h1>`
- `Element.onclick =function(){displayDate()};`
- `Element.addEventListener("click",function({});`

## DOM节点

- 添加和删除节点

- `document.createElement("p")`
- `document.createTextNode("新增")`
- `parent.appendChild(child);`
- `parent.removeChild(child);`

- ```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8" />
5          <meta name="viewport" content="width=device-width, initial-scale=1">
6          <title></title>
7      </head>
8      <body>
9          <div id="div1">
10             <p id="p1">我是第一个P</p>
11             <p id="p2">我是第二个P</p>
12         </div>
13
14         <script type="text/javascript">
15             var p = document.createElement("p");//新增<p></p>
16             var word =document.createTextNode("我是新增的P标签"); //我是新增的P
17             标签
18
19             p.appendChild(word); // <p>我是新增的P标签</p>
20
21             var div =document.getElementById("div1");
22             div.appendChild(p); //<p>我是新增的P标签</p> 放到div下
23
24             var p1 =document.getElementById("p1");
25             div.removeChild(p1); //删除p1
26         </script>
```

```
26     </body>
27 </html>
```

JS window

- window的概念
 -

window的概念

@2417065

- 所有浏览器都支持 window 对象。它表示浏览器窗口。✓
- 所有 JavaScript 全局对象、函数以及变量均自动成为 window 对象的成员。
- 全局变量是 window 对象的属性。
- 全局函数是 window 对象的方法。

- window的方法

- `window.open()` :打开新窗口
- `window.close()` :关闭当前窗口
- `window.moveTo()` :移动当前窗口
- `window.resizeTo()` :调整当前窗口的尺寸

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title></title>
7   </head>
8   <body>
9     <button onclick="openwindow()">创建窗口</button>
10    <button onclick="myFunction()">调整窗口</button>
11    <button onclick="moveFunction()">移动窗口</button>
12    <button onclick="closeFunction()">关闭窗口</button>
13
14    <script type="text/javascript">
15      var w;
16
17      function openwindow(){
18        w =window.open('','','width=300,height=300');
19      }
20
21      function myFunction(){
22
```

```

23         w.resizeTo(500,500);
24         w.focus();
25     }
26
27     function moveFunction(){
28         w.moveTo(700,500);
29         w.focus();
30     }
31
32     function closeFunction(){
33         w.close(700,500);
34         w.focus();
35     }
36     </script>
37 </body>
38 </html>

```

window.screen

-

- **window.screen** 对象在编写时可以不使用 **window** 这个前缀。
- **screen.availWidth** - 可用的屏幕宽度
- **screen.availHeight** - 可用的屏幕高度

- 可用高度不包括底边任务栏高度

- - 1 屏幕宽度和窗口宽度的区别：
 - 2 window.screen.availWidth会返回当前屏幕的宽度，不会随网页缩放而改变，在浏览器运行时固定不变。
 - 3 而window.innerWidth会返回当前窗口的宽度，会随网页缩放而改变，例如：在100%缩放时，innerHTML的值为1920；在125%缩放时，innnerWidth的值就会变成1536。但此时availWidth的值不会变，它永远等于100%缩放时的文档宽度。
 - 4
 - 5 (在windows10版本中，系统缩放125%会影响availWidth的值，即使你的屏幕是1920px宽，在系统设定125%缩放后，availWidth的值会变1536，所以availWidth并不是严格意义上的屏幕宽度，而是浏览器100%缩放时窗口的宽度。)

window Location

@2417065

- **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.protocol** 返回所使用的 web 协议 (http:// 或 https://)
- **location.href** 返回 (当前页面的) 整个 URL

`window.history`

- `window.history` 对象在编写时可不使用 `window` 这个前缀
- `history.back()` - 与在浏览器点击后退按钮相同
- `history.forward()` - 与在浏览器中点击按钮向前相同
- `history.go()`