

Js入门

2017年6月13日 15:57

上 ecmascript 规定了语法

DOM 文档对象模型

BOM 浏览器对象模型

js能做什么事情

能够让页面和浏览器与用户之间有交互行为

可与服务器进行数据交互

能动态改变页面中结构和样式

编辑器		Sublime		Atom		vscode
-----	--	---------	--	------	--	--------

1. js出现的位置

1. js写在标签行间中

```
<div onclick="alert(1)">测试</div>
```

2. 内嵌在页面中 写在一对script标签中

只能在当前的页面中使用

```
<script> </script>    type类型不需要写
```

alert 浏览器提供的函数 提醒框

```
alert(1)
```

注释：说明一段代码

单行：//

多行： /*

*

*/

3. 外链的方式

创建一个js文件，通过script标签的src链接进来

```
<script src="./test.js"></script>
```

script使用了src，写在这对script标签中的代码不执行

标签的位置：<body>结束之前 常用

<head>内

2 js操作元素

在写js代码之前，先布局

获取：

通过 id 获取元素 固定的方法
document js提供了一个操作文档的对象
document.getElementById('id名字')
id 名字一定要加上双引号或者单引号

调试：

alert() 提醒框 把要提醒的内容放在()中

需求：

点击一个按钮，弹出一个提醒框

分析：

1. 获取按钮
2. 绑定点击事件 click
3. 点击之后做一些事情
alert(1)
给按钮改变一下背景颜色/字体大小（操作标签样式）

```
<!DOCTYPE html>
<html lang="zh-cn">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  </head>
  <title></title>
  <style>
  </style>
</head>
<body>
  <input type="button" value="按钮" id="btn" />
  <script>
    document.getElementById('btn').onclick = function(){
      document.getElementById('btn').style.background = 'red'
      document.getElementById('btn').style.width = '300px'
    }
  </script>
</body>
</html>
```

3 变量

变量 存放一些数据的

要使用变量之前，先要声明变量，否则会报错 test is not defined

关键字 var

var 变量名;

声明了变量 未赋值，默认值为undefined

var 变量名 = 数据/值;

声明了变量并且赋值

使用变量就可以复用数据

= 赋值的意思 把等号右边的值赋给左边

调试代码：

chrome 调试工具

打开调试工具 f12

右击->检查

命名规则：

组成：字母、数字 _、\$组合而成

变量名不能用数字开头

不能是关键字和保留字

关键字：js正在使用的单词

var function for if break continue class ...

保留字:js即将使用的单词

参考：<http://www.runoob.com/js/js-reserved.html>

命名风格：驼峰命名法

大驼峰 多个单词组合的时候，从第一单词开始首字母大写

小驼峰 多个单词组合的时候，从第二单词开始首字母大写

起名字要有语义化

4 事件

事件

交互的一瞬间产生一个事件

在元素上会产生事件

元素.on+事件名 = function(){

事件发生之后做的事情放在这里

}

<body>

<input type="button" value="test" id="btn" />

```
<script>
    var btn = document.getElementById('btn');

    btn.onclick = function (){
        alert("1")
    };

    btn.onmouseover = function (){
        alert("2")
    };

    btn.onmouseout = function (){
        alert("3")
    };
</script>
</body>
```

5 函数

函数

把多行代码放在函数中，形成代码块

函数定义

关键字 function

语法：

```
function [函数名]() {
    //函数体/代码块
}
```

函数名遵循变量名的命名规则

[] 某些情况下代表可选的意思，可以有可以没有

函数声明：

```
function 函数名( ){
}
```

函数表达式

```
var func = function () {}
```

匿名函数

```
function () {}
```

直接定义匿名函数会报错

定义匿名函数之后立马使用，不报错

赋给一个变量

赋值给一个事件

函数调用

只定义了函数，未执行（调用），代码块不执行

函数只有调用一下才能使用
调用：函数名()

6 属性操作

向控制台打印值

`Console.log(值)`

`Console.dir(值)` 可以展开值的细节

属性

事物所具有的不可缺少的性质

操作属性：增删改查

对属性的读写操作

赋值(写)：元素.属性 = 值

取值(读)：元素.属性

. 的

[] 的

```
btn.style.width = '100px';  
btn["style"]["height"] = '200px';
```

```
var abc = 'style'
```

```
btn.abc
```

点后面的值不会解析为变量，只会认为是btn的abc这个属性

当属性值存在变量中，用[]形式

字体大小：

```
btn.style['font-size'] = '50px';
```

```
btn.style.fontSize = '50px';
```

属性名要符合变量名的命名规则

js中可以直接写一个单词，这个单词会被解析为变量，如果没有声明变量会报错
一个单词不是一个变量，只是一个值而已，加上双引号或单引号

7 style

元素.style 只能读取写在行间中的样式值

内嵌的样式值不行

下

1.元素上的属性

可读可写：

id

className

清空之前的class，重新赋值

console.log(btn.className); 没有设置class所谓为空字符串

Value

value 表单控件才有的属性

input select textarea

Input 不写type类型，默认为text；

```
<input id="txt" value="123" />
```

innerHTML

获取/设置一个元素包含的html结构

```
<body>
  <ul id="list">
    <li>123</li>
    <li>123</li>
    <li>123</li>
    <li>
      <div>
        <span>hello</span>
      </div>
    </li>
  </ul>
  <script>
    var list = document.getElementById('list');
    list.innerHTML = '<li>345</li>';
    console.log(list.innerHTML)
  </script>
</body>
```

Src/href

获取的是绝对路径

```
<body>
  
  <a href="#.html" id="lian">百度</a>
  <script>
    var imgPng = document.getElementById("imgPng");
    var lian = document.getElementById("lian");
    imgPng.src = '2.jpg'
```

```
console.log(lian.href);
```

找一个元素上不存在的属性，会返回一个值为undefined

```
</script>
```

```
</body>
```

2 留言板

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>简易留言板</title>
```

```
</head>
```

```
<body>
```

```
<input type="text" id="txt"/>
```

```
<input type="button" id="btn" value="按钮"/>
```

```
<ul id="messages">
```

```
<li>123</li>
```

```
</ul>
```

```
<script>
```

```
var txt=document.getElementById('txt');
```

```
var btn=document.getElementById('btn');
```

```
var messages=document.getElementById('messages');
```

```
btn.onclick=function(){
```

```
    messages.innerHTML=messages.innerHTML+<li>'+txt.value+'</li>;
```

```
    txt.value="";
```

```
};
```

```
</script>
```

```
</body>
```

```
</html>
```

需求：

点击按钮获取文本框的值，拼成一个结构放在ul中

```
var prevHtml = list.innerHTML;
```

```
console.log(prevHtml);
```

想保留之前的结构，需要把之前的结构和最新的结构拼接起来

```
list.innerHTML = '<li>'+message.value+'</li>' + prevHtml
```

3 字符串拼接

字符串：

由一对单引号或双引号包起来的0个或多个字符组成的串

'123' "abc"

" 空字符串

'空格' 有值的

字符串拼接

+ 来做字符串拼接

1. 加号的左边和右边都是数字，做加法运算得到结果
2. 加号的一边有一个是字符串，就是字符串拼接，不是字符串的一边转成字符串。

```
<body>
  <script>
    alert(3+4)
    alert('3'+1)
  </script>
</body>
```