

JavaScript 辅助讲义

一、DOM

1.DOM 的简述

DOM 全称 Document Object Model, 即文档对象模型, 它允许脚本(js)控制 Web 页面、窗口和文档。

想要做一个动态页面, 需要交互。这时, 静态页面是无法满足的, 比如你在博客里, 修改个人信息, 添加一个标签页, 等等都需要同步的额。

2. DOM 基本功能

- 查询某个元素
- 查询某个元素的祖先、兄弟以及后代元素
- 获取、修改元素的属性
- 获取、修改元素的内容
- 创建、插入和删除元素

3. DOM 的获取步骤

1. 在浏览器加载一个页面时, 浏览器会解析 HTML, 并创建文档的一个内部模型, 其中包含 HTML 标记的所有元素, 自上而下解析, 遇到 JavaScript 浏览器会检查它的正确性, 然后执行代码。
2. JavaScript 继续执行, 使用 DOM 检查页面、完成修改、从页面接受事件, 或者要求浏览器从 Web 服务器获取其它数据 注: document 是一个反映 HTML 的对象, 通过调用 document 的方法改变 DOM 的状态, 也就是改变 HTML 页面
3. JavaScript 修改了 DOM 时, 浏览器会随着动态更新页面。

二、元素的操作（应用部分）

1.document——获取元素

document 接口表示任何在浏览器中已经加载好的网页, 并作为一个入口去操作网页内容。

它提供了全局操作 document 的功能, 像获取网页的 URL 和在 document 里创建一个新的元素。

`document.getElementById()` 在文档中按照 id 属性查找元素参数写 id 名。

```
//获取到html元素中id叫做myDiv的元素整体
var myDiv = document.getElementById('myDiv');
```

2.事件——与元素交互

可以在 DOM 元素上绑定 `onclick`、`onmouseover`、`onmouseout`、`onmousedown`、`onmouseup`、`ondblclick`、`onkeydown`、`onkeyup` 等。JavaScript 能够在事件发生时执行。

(1) 在 DOM 中直接绑定事件

直接在标签中设置事件属性, 绑定方法, 从而完成交互。

```
<input type="button" value="点我" onclick="hello(1)" />
<script>
  function hello(a) {
    alert('hello world!');
    // alert(a);
  }
</script>
```

(2) 在 JavaScript 代码中绑定事件

可以通过 js 获取元素, 在 js 中绑定获取的元素相关的事件, 调用函数处理事件功能。

```
var btn = document.getElementById('mybtn');
btn.onclick = function () {
    alert('hello');
    //其他语句, 包括调用函数语句如下:
    //abc()
};
```

(3) 在JavaScript通过事件调用函数

一个命名函数, 当元素事件触发可以直接调用这个函数的名字, 不需要在函数后增加括号, 如果增加括号, 函数没有等点击触发就会执行。

```
var btn = document.getElementById('mybtn');
btn.onclick = abc;
function abc() {
    alert('hello');
};
```

2.Element——innerHTML

Element.innerHTML 获取/设置元素的内容。innerHTML 在 JS 是双向功能: 获取对象的内容或向对象插入内容。

```
<div id="myDiv">div元素</div>
<script>
    var myDiv = document.getElementById('myDiv');
    myDiv.innerHTML = '<p>一个元素</p>';
</script>
```

- 可以通过 `document.getElementById("myDiv").innerHTML` 来获取 id 为 `myDiv` 的对象的内嵌内容;
- 也可以对某对象插入内容, 如 `document.getElementById("myDiv").innerHTML='这是被插入的内容';` 这样就能向 id 为 `myDiv` 的对象插入内容。

3. Element——value

元素的 value 属性可以直接获取到具有 value 属性元素的值。多数情况用在获取表单相关元素中。

```
<input type="text" id="nn" />
<input type="button" value="获取" onclick="print()" />
<script>
    var a = document.getElementById('nn');
    //通过事件获取值
    function print() {
        alert(a.value);
    }
</script>
```

练习

在页面上放入一个文本输入框、一个密码输入框、一个按钮 使用 js 获取两个输入框元素和一个按钮元素 点击按钮元素, 触发点击事件, 获得文本输入框和密码输入框的内容, 并在控制台查看 创建两个 p 标签, 使用 js 获取两个 p 元素 将获取的用户输入的用户名和密码显示页面上

4. 清空多余空格

(1) trim()方法

- `trim()` 方法用于删除字符串的"头尾"空白符, 空白符包括: 空格、制表符 tab、换行符等其他空白符等。
- `trim()` 方法不会改变原始字符串。
- `trim()` 方法不适用于 null, undefined, Number 类型。

```
var str = ' hello ';
alert(str.trim());
```

非空验证

```
var str1 = '   ';
var str2 = ''
var str3 = ' tom ';
function notEmpty(s){
    var nstr = s.trim();
    return nstr.length;
}
notEmpty(str1)
notEmpty(str2)
notEmpty(str3)
```

(2) replace()方法

使用字符串的 `replace` 方法参数放入正则，可以将空格替换成空，以达到清除空格的目的。

方法一：

```
var str1 = '你好 我 是 t o m';
str1 = str1.replace(/[ ]/g, ''); //替换所有空格!
console.log(str1);
```

方法二：

- 去除字符串内所有的空格：`str = str.replace(/\s*/g, "");`
- 去除字符串内两头的空格：`str = str.replace(/^\s*|\s*$/g, "");`
- 去除字符串内左侧的空格：`str = str.replace(/^\s*/, "");`
- 去除字符串内右侧的空格：`str = str.replace(/\s*$/g, "");`

```
var str2 = '当然 我 是 j a c k';
str2 = str2.replace(/\s*/g, ''); //替换所有空格!
console.log(str2);
```

练习

前台的非空验证 如果用户输入的内容为空，那么在前台作出提示

三、JSON

1. JSON 简述

JSON(JavaScript Object Notation, JS 对象简谱) 是一种轻量级的数据交换格式。它基于 ECMAScript (欧洲计算机协会制定的 js 规范)的一个子集，采用完全独立于编程语言的文本格式来存储和表示数据。

2. JSON 语法规则

JSON 是一个序列化的对象或数组。属性名 (key) 用双引号包裹，值可以是对象、数组、数字、字符串或者三个字面值(false、null、true)中的一个。值中的字面值中的英文必须使用小写。

(1) 对象结构

```
{
  "key1": "value1",
  "key2": "value2"
}
```

(2) 数组结构

```
[
  {
    "key1": "value1",
    "key2": "value2"
  },
  {
    "key3": "value3",
    "key4": "value4"
  }
]
```

2. JSON和JS对象的关系

JSON是 JS的一种简单数据格式，JSON是JavaScript原生格式，它是一种严格的JS对象的格式，JSON的属性名必须有双引号，如果值是字符串，也必须是双引号。

区别	JSON	Javascript
含义	仅仅是一种数据格式	表示类的实例
传输	可以跨平台数据传输，速度快	不能传输
表现	1.简直对方式，键必须加双引号	1.键值对方式，键不加引号
	2.值不能是方法函数，不能是undefined/NaN	2.值可以是函数、对象、字符串、数字、boolean 等
相互转换	Json转换Js对象	js对象转换Json
	1.JSON.parse(JsonStr);(不兼容IE7)	JSON.stringify(jsObj);
	2.eval("(" + jsonStr + ")");(兼容所有浏览器，但不安全，会执行json里面的表达式)	
其他	调用JSON官网的JS,实现parse和stringify在哥哥浏览器的兼容：	

其中，eval()函数是自行参数内，字符串的js代码，如:eval("2+2")

2. JSON字符串和JSON对象的区别

JSON字符串：指的是符合JSON格式要求的JS字符串。需要注意的是，服务器的返回数据，以及前台向后台的传输数据都可能是JSON字符串，而不是JSON对象。

```
var jsonStr = "{id:'001',name:'tom',age:18}";
```

JSON对象：指符合JSON格式要求的JS对象。

```
var jsonObj = { id: '001', name: 'tom', age: 18 };
```

3. 转换方式

为什么要使用JSON的转化，因为前后台数据的传输均为JSON字符串格式，如不通过解析和转化将无法使用。

JSON.stringify() 用于将一个值转为字符串。该字符串应该符合JSON格式，并且可以被 JSON.parse 方法还原。

```
var obj = JSON.parse('{ "a": "Hello", "b": "World" }');
//结果是 {a: 'Hello', b: 'World'}
```

JSON.parse() 用于将JSON字符串转化成对象。

```
var json = JSON.stringify({ a: 'Hello', b: 'World' });
//结果是 '{ "a": "Hello", "b": "World" }'
```

练习

使用JS文件模拟数据当作后台返回的数据 使用遍历的方式将数据展现在前台（innerHTML） join('') 标签拼接的时候去掉空格，例

如： tab.innerHTML = list.join(''); 真实数据接

□: https://movie.douban.com/j/search_subjects?type=movie&tag=%E7%83%AD%E9%97%A8&sort=recommend&page_limit=10&page_start=0

```
var json = {
  "subjects": [
    {
      "rate": "6.1",
      "title": "盖亚",
      "url": "https://movie.douban.com/subject/35363988/",
      "cover": "https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2654849860.webp",
      "id": "35363988",
      "is_new": true
    },
    {
      "rate": "7.3",
      "title": "夏日友晴天",
      "url": "https://movie.douban.com/subject/35161768/",
      "cover": "https://img1.doubanio.com/view/photo/s_ratio_poster/public/p2633883449.webp",
      "id": "35161768",
      "is_new": false
    },
    {
      "rate": "7.6",
      "title": "悬崖之上",
      "url": "https://movie.douban.com/subject/32493124/",
      "cover": "https://img2.doubanio.com/view/photo/s_ratio_poster/public/p2641614433.webp",
      "id": "32493124",
      "is_new": false
    },
    {
      "rate": "7.5",
      "title": "浪客剑心 最终章 人诛篇",
      "url": "https://movie.douban.com/subject/30167974/",
      "playable": false,
      "cover": "https://img2.doubanio.com/view/photo/s_ratio_poster/public/p2627551613.webp",
      "id": "30167974",
      "is_new": false
    },
    {
      "rate": "7.0",
      "title": "黑白魔女库伊拉",
      "url": "https://movie.douban.com/subject/26703121/",
      "cover": "https://img9.doubanio.com/view/photo/s_ratio_poster/public/p2653946775.webp",
      "id": "26703121",
      "is_new": false
    }
  ]
}
```