

day01 - Web APIs


学习目标：

能够通过ID来获取元素 能够通过标签名来获取元素 能够通过class来获取元素 能够通过选择器来获取元素 能够获取body和html元素 能够给元素注册事件 能够修改元素的内容 能够区分innerText和innerHTML的区别 能够修改像div这类普通元素的属性 能够修改表单元素的属性 能够修改元素的样式属性

1.1. Web API介绍

1.1.1 API的概念

API (Application Programming Interface , 应用程序编程接口) 是一些预先定义的函数, 目的是提供应用程序与开发人员基于某软件或硬件得以访问一组例程的能力, 而又无需访问源码, 无需理解其内部工作机制细节, 只需直接调用使用即可。

1550719355829

举例解释什么是API。

例如,

C语言中有一个函数 `fopen()` 可以打开硬盘上的文件, 这个函数对于我们来说, 就是一个C语言提供的打开文件的工具。

javascript中有一个函数 `alert()` 可以在页面弹一个提示框, 这个函数就是js提供的一个弹框工具。

这些工具 (函数) 由编程语言提供, 内部的实现已经封装好了, 我们只要学会灵活的使用这些工具即可。

1.1.2 Web API的概念

Web API 是浏览器提供的一套操作浏览器功能和页面元素的 API (BOM 和 DOM)。

现阶段我们主要针对于浏览器讲解常用的 API, 主要针对浏览器做交互效果。比如我们想要浏览器弹出一个警示框, 直接使用 `alert('弹出')`

MDN 详细 API : <https://developer.mozilla.org/zh-CN/docs/Web/API>

因为 Web API 很多, 所以我们将这个阶段称为 Web APIs。

此处的 Web API 特指浏览器提供的一系列API(很多函数或对象方法), 即操作网页的一系列工具。例如: 操作html标签、操作页面地址的方法。

1.1.3 API 和 Web API 总结

1. API 是为我们程序员提供的一个接口, 帮助我们实现某种功能, 我们会使用就可以了, 不必纠结内部如何实现
2. Web API 主要是针对于浏览器提供的接口, 主要针对于浏览器做交互效果。

3. Web API 一般都有输入和输出（函数的传参和返回值），Web API 很多都是方法（函数）
4. 学习 Web API 可以结合前面学习内置对象方法的思路学习

1.2. DOM 介绍

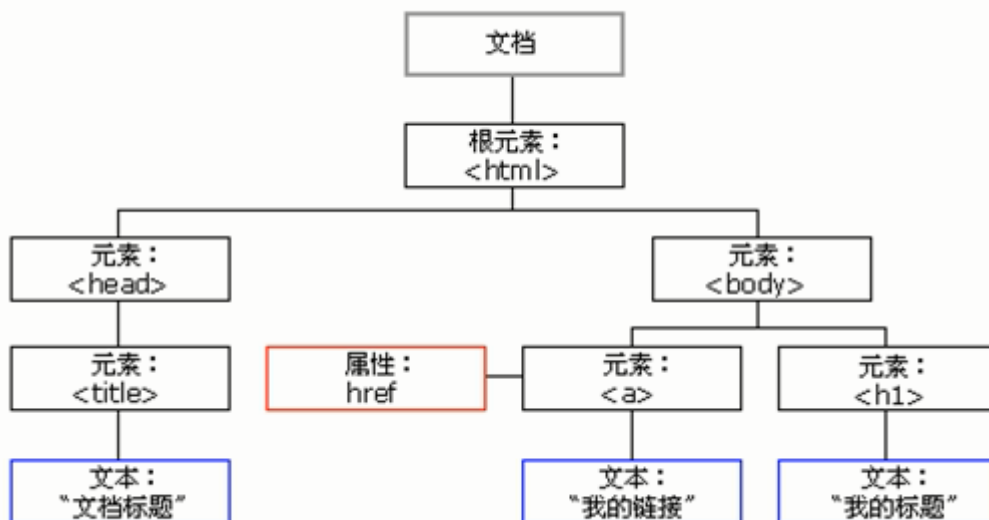
1.2.1 什么是DOM

文档对象模型（Document Object Model，简称DOM），是 [W3C](#) 组织推荐的处理[可扩展标记语言](#)（html或者xhtml）的标准[编程接口](#)。

W3C 已经定义了一系列的 DOM 接口，通过这些 DOM 接口可以改变网页的内容、结构和样式。

DOM是W3C组织制定的一套处理 html和xml文档的规范，所有的浏览器都遵循了这套标准。

1.2.2. DOM树



DOM树 又称为文档树模型，把文档映射成树形结构，通过节点对象对其处理，处理的结果可以加入到当前的页面。

- 文档：一个页面就是一个文档，DOM中使用document表示
- 节点：网页中的所有内容，在文档树中都是节点（标签、属性、文本、注释等），使用node表示
- 标签节点：网页中的所有标签，通常称为元素节点，又简称为“元素”，使用element表示

DOM 把以上内容都看做是对象

1.3. 获取元素

为什么要获取页面元素？

例如：我们想要操作页面上的某部分(显示/隐藏，动画)，需要先获取到该部分对应的元素，再对其进行操作。

1.3.1. 根据ID获取

语法：document.getElementById(id)
作用：根据ID获取元素对象
参数：id值，区分大小写的字符串
返回值：元素对象 或 null

案例代码

```
<body>
  <div id="time">2019-9-9</div>
  <script>
    // 因为我们文档页面从上往下加载，所以先得有标签 所以我们script写到标签的下面
    var timer = document.getElementById('time');
    console.log(timer);
    console.log(typeof timer);
    // console.dir 打印我们返回的元素对象 更好的查看里面的属性和方法
    console.dir(timer);
  </script>
</body>
```

1.3.2. 根据标签名获取元素

语法：document.getElementsByTagName('标签名') 或者 element.getElementsByTagName('标签名')
作用：根据标签名获取元素对象
参数：标签名
返回值：元素对象集合（伪数组，数组元素是元素对象）

案例代码

```
<body>
  <ul>
    <li>知否知否，应是等你好久11</li>
    <li>知否知否，应是等你好久22</li>
    <li>知否知否，应是等你好久33</li>
    <li>知否知否，应是等你好久44</li>
    <li>知否知否，应是等你好久55</li>
  </ul>
  <ul id="nav">
    <li>生僻字</li>
    <li>生僻字</li>
    <li>生僻字</li>
    <li>生僻字</li>
    <li>生僻字</li>
  </ul>
  <script>
    // 1. 返回的是 获取过来元素对象的集合 以伪数组的形式存储的
    var lis = document.getElementsByTagName('li');
    console.log(lis);
    console.log(lis[0]);
    // 2. 我们想要依次打印里面的元素对象我们可以采取遍历的方式
    for (var i = 0; i < lis.length; i++) {
      console.log(lis[i]);
    }
  </script>
```

```

    }
    // 3. element.getElementsByTagName() 可以得到这个元素里面的某些标签
    var nav = document.getElementById('nav'); // 这个获得nav 元素
    var navLis = nav.getElementsByTagName('li');
    console.log(navLis);
</script>
</body>

```

注意:

1. 因为得到的是一个对象的集合，所以我们想要操作里面的元素就需要遍历。
2. 得到元素对象是动态的

注意：getElementsByTagName()获取到是动态集合，即：当页面增加了标签，这个集合中也就增加了元素。

1.3.3. H5新增获取元素方式

```
1. document.getElementsByClassName('类名'); // 根据类名返回元素对象集合
```

```
2. document.querySelector('选择器'); // 根据指定选择器返回第一个元素对象
```

```
3. document.querySelectorAll('选择器'); // 根据指定选择器返回
```

注意:

querySelector 和 querySelectorAll里面的选择器需要加符号,比如:document.querySelector('#nav');

案例代码

```

<body>
  <div class="box">盒子1</div>
  <div class="box">盒子2</div>
  <div id="nav">
    <ul>
      <li>首页</li>
      <li>产品</li>
    </ul>
  </div>
<script>
  // 1. getElementsByClassName 根据类名获得某些元素集合
  var boxs = document.getElementsByClassName('box');
  console.log(boxs);
  // 2. querySelector 返回指定选择器的第一个元素对象 切记 里面的选择器需要加符号 .box #nav
  var firstBox = document.querySelector('.box');
  console.log(firstBox);
  var nav = document.querySelector('#nav');

```

```
console.log(nav);
var li = document.querySelector('li');
console.log(li);
// 3. querySelectorAll() 返回指定选择器的所有元素对象集合
var allBox = document.querySelectorAll('.box');
console.log(allBox);
var lis = document.querySelectorAll('li');
console.log(lis);

</script>
</body>
```

1.3.4 获取特殊元素 (body , html)

获取body元素

```
1. document.body // 返回body元素对象
```

获取html元素

```
1. document.documentElement // 返回html元素对象
```

热门课程，等你来选

北美大数据

JAVA架构师

WEB工程师

网络营销师

UI/UE设计师

人工智能

大家如果有亲属、同学以及朋友有培训需求的请联系我

