

WEB API

web API的介绍

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, 或理解内部工作机制的细节。

- 任何开发语言都有自己的API
- API的特征输入和输出(I/O)
 - 内置对象的方法都是API
 - `var max = Math.max(1,2,3)`
- API的使用方法(`console.log()`)

Web API

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)

此处的Web API特指浏览器提供的API(一组方法), Web API在后面的课程中有其它含义

掌握常见的浏览器提供的API的调用方式

[MDN-Web API](#)

JavaScript的组成

JavaScript: 包含 ECMAScript、DOM(Document Object Model)、BOM(Browser Object Model)

ECMAScript

- JavaScript的核心
- 描述了语言的基本语法和数据类型, ECMAScript是一套标准, 定义了一种语言的标准与具体实现无关

BOM

- 浏览器对象模型
- 一套操作浏览器功能的API
- 通过BOM可以操作浏览器窗口, 比如: 弹出框、控制浏览器跳转、获取分辨率等

DOM

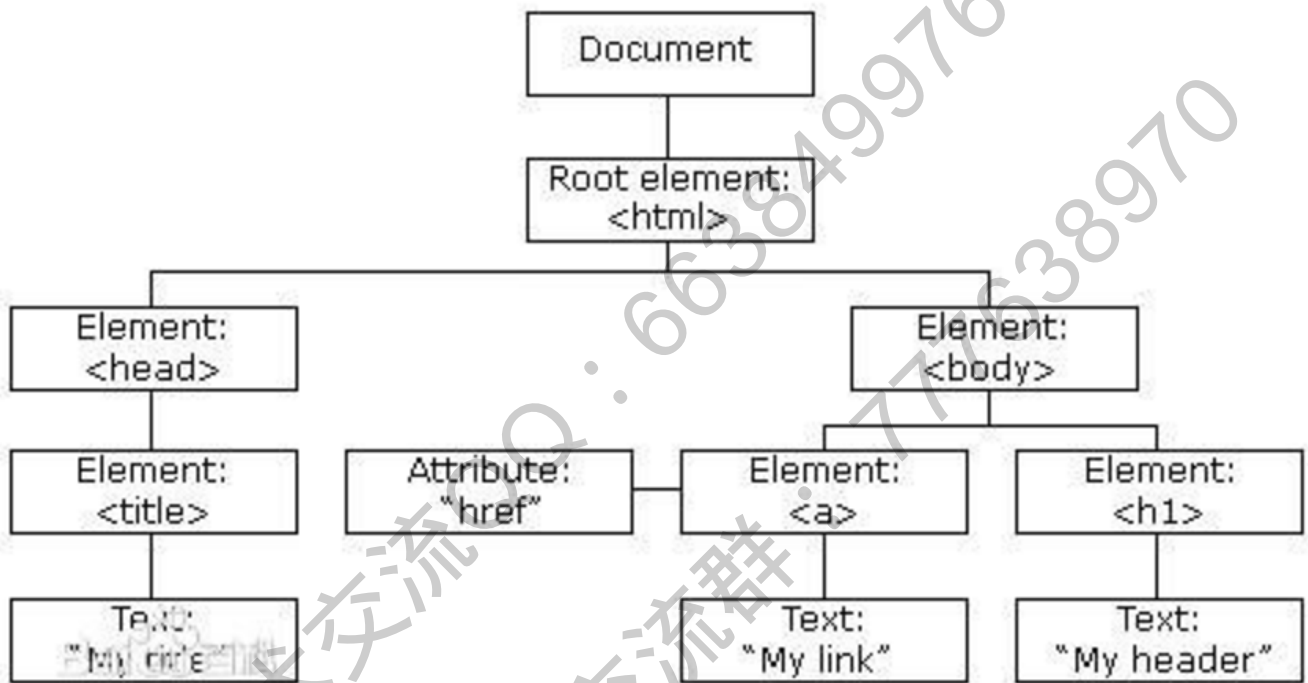
- 文档对象模型
- 一套操作页面元素的API
- DOM可以把HTML看做是文档树, 通过DOM提供的API可以对树上的节点进行操作

文档对象模型

DOM的概念

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志语言的标准编程接口。在网页上，组织页面（或文档）的对象被组织在一个树形结构中，用来表示文档中对象的标准模型就称为DOM。Document Object Model的历史可以追溯至1990年代后期微软与Netscape的“浏览器大战”，双方为了在JavaScript与JScript一决生死，于是大规模的赋予浏览器强大的功能。微软在网页技术上加入了不少专属事物，既有VBScript、ActiveX、以及微软自家的DHTML格式等，使不少网页使用非微软平台及浏览器无法正常显示。DOM即是当时蕴酿出来的杰作。

DOM又称为文档树模型



- 文档：一个网页可以称为文档
- 节点：网页中的所有内容都是节点（标签、属性、文本、注释等）
- 元素：网页中的标签
- 属性：标签的属性

DOM经常进行的操作

- 获取元素
- 动态创建元素
- 对元素进行操作(设置其属性或调用其方法)
- 事件(什么时机做相应的操作)

获取页面元素

为什么要获取页面元素

我们想要操作页面上某个部分（显示/隐藏，动画），需要先获得该部分对应的元素，才能进行后续的操作。

根据id获取元素

```
1 var div = document.getElementById('main');
2 console.log(div);
3
4 // 获取到的数据类型 HTMLDivElement, 对象都是有类型的
5 // HTMLDivElement <-- HTMLElement <-- Element <-- Node <-- EventTarget
```

注意：由于id名具有唯一性，部分浏览器支持直接使用id名访问元素，但不是标准方式，不推荐使用。

根据标签名获取元素

```
1 var divs = document.getElementsByTagName('div');
2 for (var i = 0; i < divs.length; i++) {
3     var div = divs[i];
4     console.log(div);
5 }
```

注意：函数名为复数形式，返回值，也是复数

以上两个方法，一定要掌握，没有任何兼容性，所有浏览器都支持
其它的获取元素的方法，可能会存在兼容性问题

根据name获取元素*

```
1 var inputs = document.getElementsByName('hobby');
2 for (var i = 0; i < inputs.length; i++) {
3     var input = inputs[i];
4     console.log(input);
5 }
```

根据类名获取元素

```
1 var mains = document.getElementsByClassName('main');
2 for (var i = 0; i < mains.length; i++) {
3     var main = mains[i];
4     console.log(main);
5 }
```

根据选择器获取元素

```
1 var text = document.querySelector('#text');
2 console.log(text);
3
4 var boxes = document.querySelectorAll('.box');
5 for (var i = 0; i < boxes.length; i++) {
6     var box = boxes[i];
7     console.log(box);
8 }
```

总结

掌握

1 | 1. getElementById

2. getElementsByTagName

了解

1 | 1. getElementsByName

2. getElementsByClassName

3. querySelector

4. querySelectorAll

作业:

上课案例全部敲一遍，特别是没有基础的