Web API简介

了解 Web API 的内容

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。

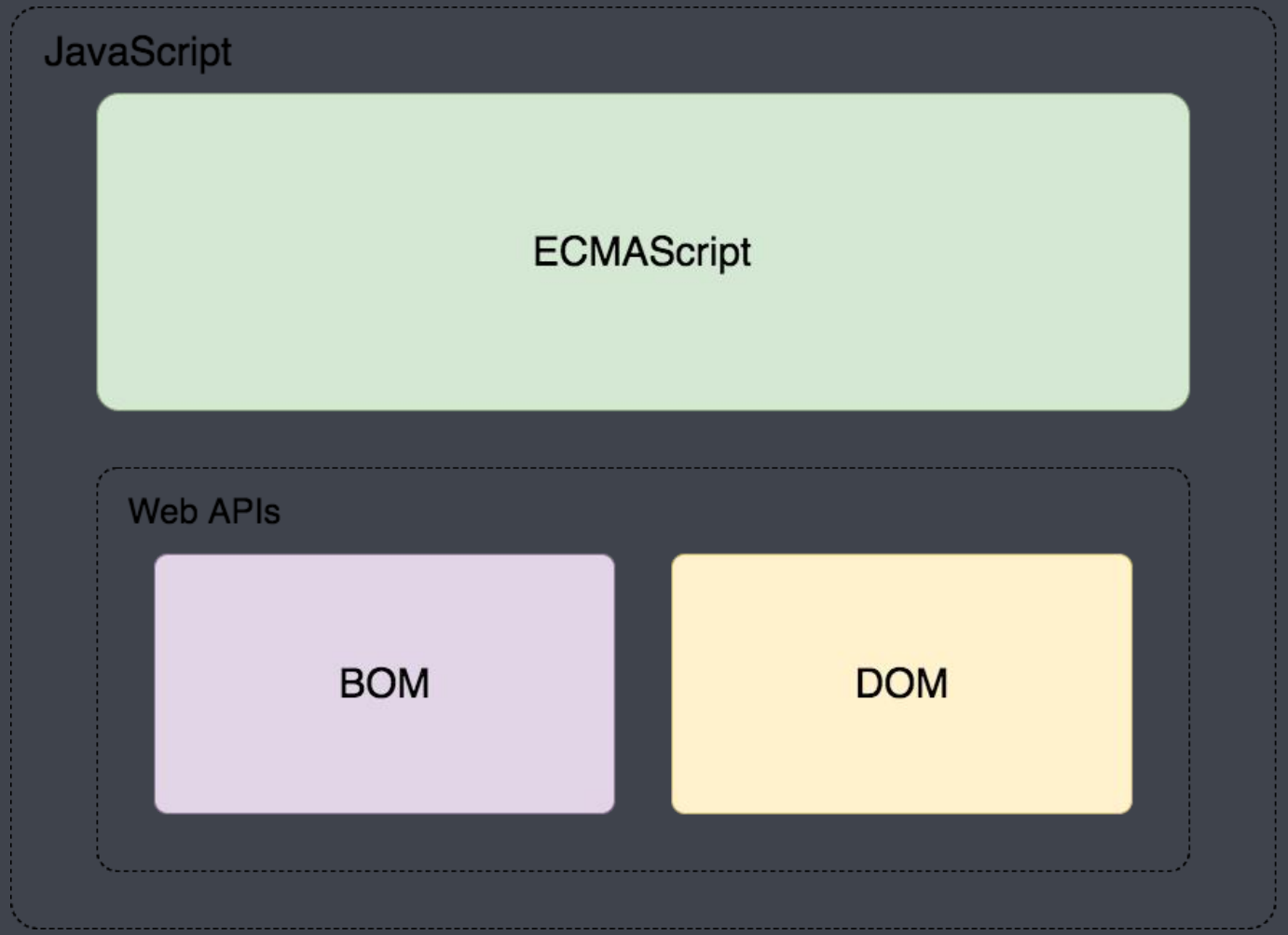
- 任何开发语言都有自己的 API
- API 的特征输入和输出(I/O)

  例如：var max = Math.max(2,3,4);
- API 的使用方法(console.log("abc"))

Web API 的概念

- 浏览器提供了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。
- 学习目标：掌握常见浏览器提供的 API 的调用方式。
- 学习辅助 MDN：https://developer.mozilla.org/zh-CN/docs/Web/API

JavaScript 的组成



ECMAScript - JavaScript 的核心

- 定义了 JavaScript 的语法规范。
- 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

- browser object model，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

- document object model，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。

DOM 简介

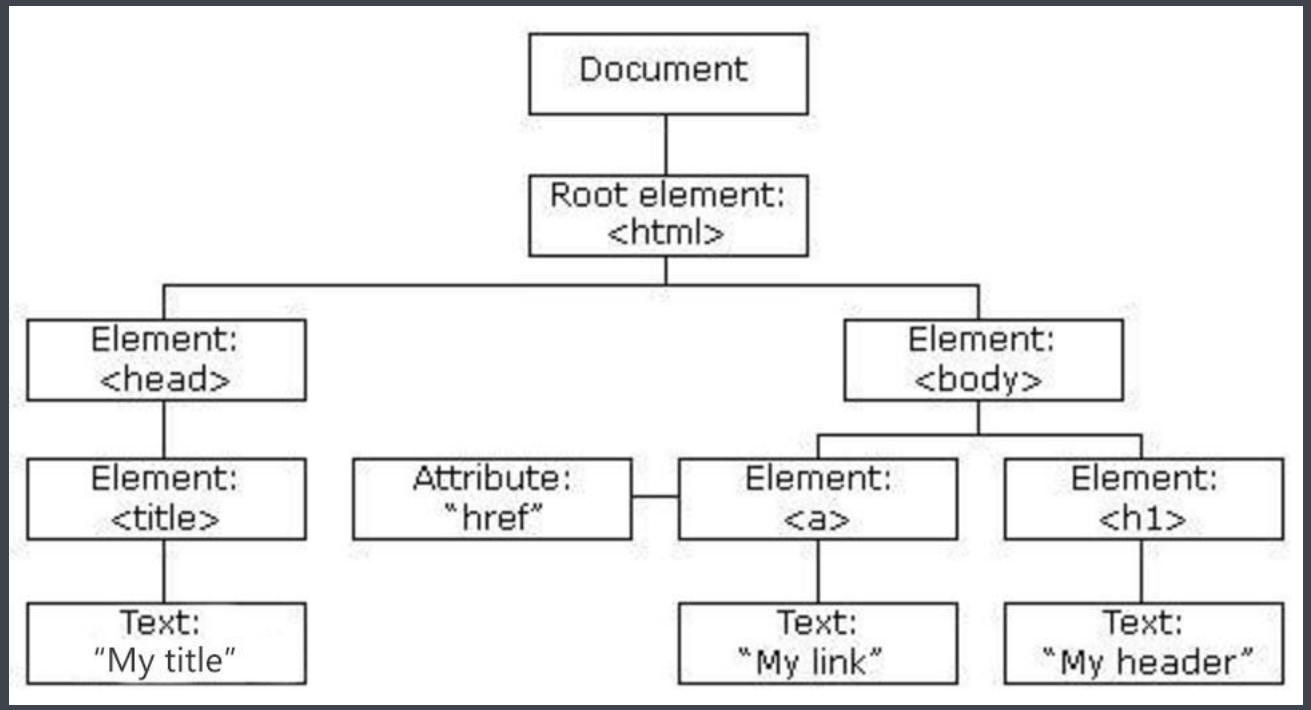
了解 DOM 的概念

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记语言的标准编程接口。它是一种与平台和语言无关的应用程序接口(API)，它可以动态地访问程序和脚本，更新其内容、结构和 www 文档的风格(目前，HTML 和 XML 文档是通过说明部分定义的)。文档可以进一步被处理，处理的结果可以加入到当前的页面。DOM 是一种基于树的 API 文档，它要求在处理过程中整个文档都表示在存储器中。

DOM 树

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

DOM 获取页面元素

了解 DOM 获取元素的各种方法

获取页面元素

- 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，才进行后续操作。

根据 id 获取元素

- 方法：调用 document 对象的 getElementById 方法。
- 参数：字符串类型的 id 的属性值。
- 返回值：对应 id 名的元素对象。
- 注意1：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。
- 注意2：代码执行顺序，如果 js 在 html 结构之前，会导致结构未加载，不能获取对应id的元素。

根据标签名获取元素

- 方法：调用 document 对象的 getElementsByTagName 方法。
- 参数：字符串类型的标签名。
- 返回值：同名的元素对象组成的数组。
- 注意1：操作数据时需要按照操作数组的方法进行。
- 注意2：getElementsByTagName 方法内部获取的元素是动态增加的。

元素对象内部获取标签元素

- 获取的元素对象内部，本身也可以调用根据标签获取元素方法，例如 div 元素对象也可以调用 getElementsByTagName 方法。
- 目的：缩小选择元素的范围，类似 css 中的后代选择器。

根据 name 获取元素

- 方法：调用 document 对象的 getElementsByName 方法。
- 参数：字符串类型的 name 属性值。
- 返回值：name 属性值相同的元素对象组成的数组。
- 不建议使用：在 IE 和 Opera 中有兼容问题，会多选中 id 属性值相同的元素。

根据类名获取元素

- 方法：调用 document 对象的 getElementsByClassName 方法。
- 参数：字符串类型的 class 属性值。
- 返回值：class 属性值相同的元素对象组成的数组。
- 浏览器兼容问题：不支持 IE8 及以下的浏览器

根据选择器获取元素

- 方法1：调用 document 对象的 querySelector 方法，通过 css 中的选择器去选取第一个符合条件的标签元素。
- 方法2：调用 document 对象的 querySelectorAll 方法，通过 css 中的选择器去选取所有符合条件的标签元素。
- 参数：字符串类型的 css 中的选择器。
- 浏览器兼容问题：不支持 IE8 以下的浏览器

总结

掌握，没有兼容问题

- getElementById()
- getElementsByTagName()

了解

- getElementsByName()
- getElementsByClassName()
- querySelector()
- querySelectorAll()

DOM 事件基本应用

掌握 DOM 事件的用法

事件

- 事件：在什么时候做什么事
- 执行机制：触发--响应机制
- 绑定事件(注册事件)三要素：
    1、事件源：给谁绑定事件
    2、事件类型：绑定什么类型的事件 click 单击
    3、事件函数：事件发生后执行什么内容，写在函数内部

事件监听

- JavaScript 解析器会给有绑定事件的元素添加一个监听，解析器会一直监测这个元素，只要触发对应的绑定事件，会立刻执行事件函数。

常用事件监听方法

- 方法1：绑定 HTML 元素属性。
- 方法2：绑定 DOM 对象属性。

常用的鼠标事件类型

- onclick　　鼠标左键单击触发
- ondbclick　　鼠标左键双击触发
- onmousedown　鼠标按键按下触发
- onmouseup　　鼠标按键放开时触发
- onmousemove　鼠标在元素上移动触发
- onmouseover　鼠标移动到元素上触发
- onmouseout　　鼠标移出元素边界触发

案例

- 点击按钮弹出提示框

DOM 元素属性操作

掌握 DOM 元素属性操作方式

非表单元素的属性

- 例如：href、title、id、src 等。
- 调用方式：元素对象打点调用属性名，例如 obj.href。
- 注意：部分的属性名跟关键字和保留字冲突，会更换写法。

  class → className

  for → htmlFor

  rowspan → rowSpan
- 属性赋值：给元素属性赋值,等号右侧的赋值都是字符串格式。

案例

- 点击按钮切换图片
- 点击按钮显示隐藏 div
- 美女相册

获取标签内部内容的属性

- 获取标签内部内容的属性有两个：innerHTML 和 innerText
- innerHTML属性，在获取标签内部内容时，如果包含标签，获取的内容会包含标签，获取的内容包括空白换行等。
- innerText属性，在获取标签内部内容时，如果包含标签，获取的内容会过滤标签，获取的内容会去掉换行和缩进等空白。

更改标签内容

还可以通过两个属性给双标签内部去更改内容：

- innerHTML 设置属性值，有标签的字符串，会按照 HTML 语法中的标签加载。
- innerText 设置属性值，有标签的字符串，会按照普通的字符加载。

对比使用场景

- innerText：在设置纯字符串时使用。
- innerHTML：在设置有内部子标签结构时使用。

表单元素属性

- value 用于大部分表单元素的内容获取(option除外)
- type 可以获取input标签的类型(输入框或复选框等)
- disabled 禁用属性
- checked 复选框选中属性
- selected 下拉菜单选中属性
- 注意：在 DOM 中元素对象的属性值只有一个时，会被转成布尔值显示。

  例如：txt.disabled = true;

案例

- 检测用户名是否是3-6位，密码是否是6-8位，如果不满足要求高亮显示文本框
- 设置下拉框中的选中项
- 搜索文本框
- 全选反选

自定义属性操作

- getAttribute(name) 获取标签行内属性
- setAttribute(name, value) 设置标签行内属性
- removeAttribute(name) 移除标签行内属性
- 与element.属性的区别：上述三个方法用于获取任意的行内属性，包括自定义的属性。

style 样式属性操作

- 使用 style 属性方式设置的样式显示在标签行内。
- element.style 属性的值，是所有**行内样式**组成的一个样式对象。
- 样式对象可以继续点语法调用或更改 css 的行内样式属性，例如 width、height 等属性。
- 注意1：类似 background-color 这种复合属性的单一属性写法，是由多个单词组成的，要修改为驼峰命名方式书写 backgroundColor。
- 注意2：通过样式属性设置宽高、位置的属性类型是字符串，需要加上 px 等单位。

className 类名属性操作

- 修改元素的 className 属性相当于直接修改标签的类名。
- 如果需要修改多条 css 样式，可以提前将修改后的样式设置到一个类选择器中，后续通过修改类名的方式，批量修改 css 样式。

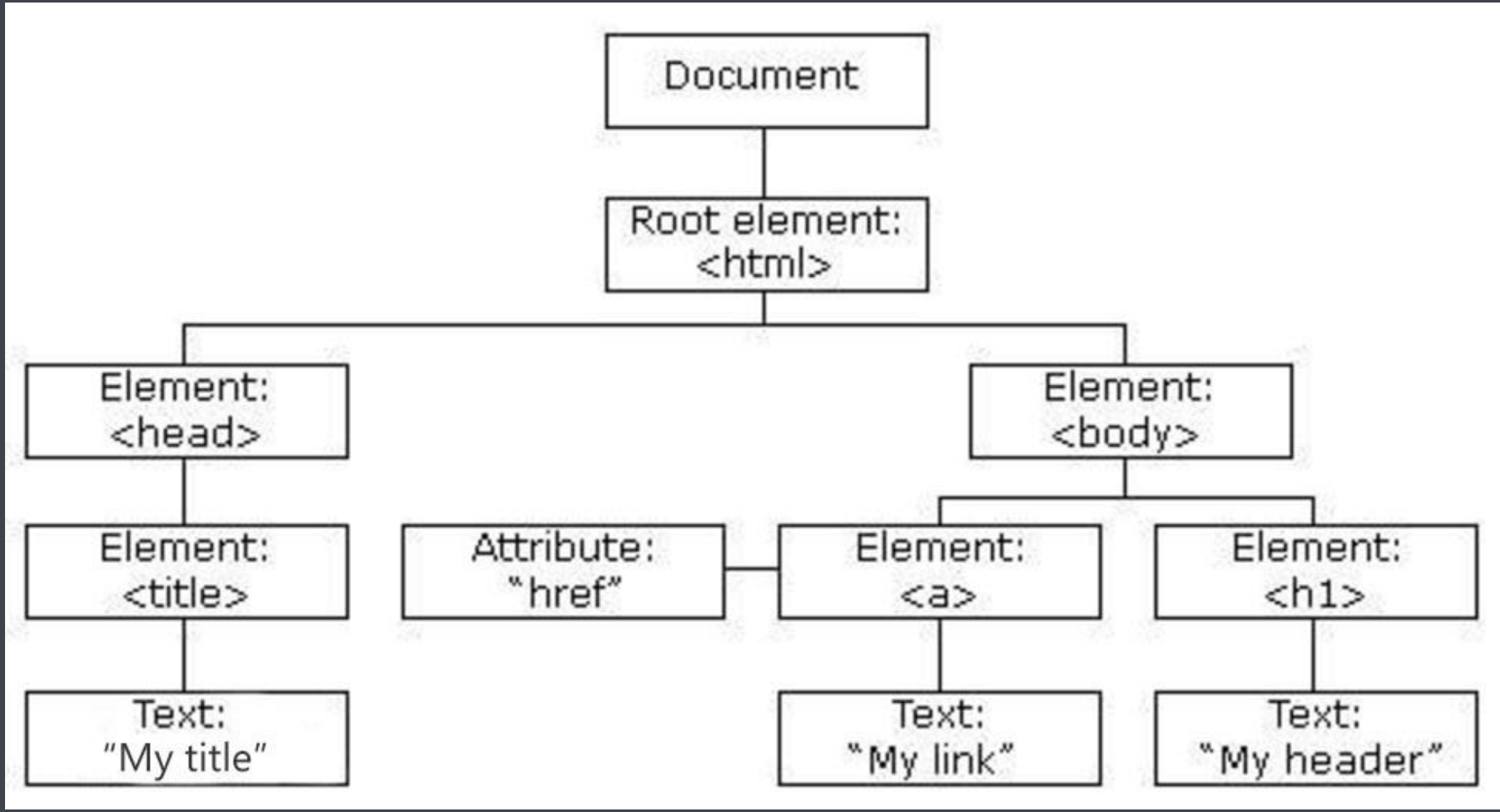
案例

- 开关灯
- 图片切换二维码案例
- 当前输入的文本框高亮显示
- 点击按钮改变div的大小和位置
- 表格隔行变色、高亮显示
- tab选项卡切换

DOM 节点操作

掌握 DOM 节点操作方法

DOM 树


Document
Root element:
<html>
Element:
<head>
Element:
<body>
Element:
<title>
Attribute:
"href"
Element:
<a>
Element:
<h1>
Text:
"My title"
Text:
"My link"
Text:
"My header"

节点属性

- nodeType　节点的类型，属性值为数字，表示不同的节点类型，共 12 种，只读

  1 元素节点

  2 属性节点

  3 文本节点

- nodeName　节点的名称(标签名称)，只读
- nodeValue　节点值，返回或设置当前节点的值

  元素节点的 nodeValue 始终是 null

父子节点常用属性

- childNodes　只读属性，获取一个节点所有子节点的实时的集合，集合是动态变化的。
- children　只读属性，返回一个节点所有的子元素节点集合，是一个动态更新的 HTML 元素集合。
- firstChild　只读属性，返回该节点的第一个子节点，如果该节点没有子节点则返回 null。
- lastChild　只读属性，返回该节点的最后一个子节点，如果该节点没有子节点则返回 null。
- parentNode　返回一个当前节点的父节点，如果没有这样的节点，比如说像这个节点是树结构的顶端或者没有插入一棵树中，这个属性返回 null。
- parentElement 返回当前节点的父元素节点，如果该元素没有父节点，或者父节点不是一个 DOM 元素，则返回 null。

兄弟节点常用属性

- nextSibling　　只读属性，返回与该节点同级的下一个节点，如果没有返回null。
- previousSibling　　只读属性，返回与该节点同级的上一个节点，如果没有返回null。
- nextElementSibling　只读属性，返回与该节点同级的下一个元素节点，如果没有返回null。
- previousElementSibling 只读属性，返回与该节点同级的上一个元素节点，如果没有返回null。
- 注意：nextElementSibling 和 previousElementSibling 有兼容性问题，IE9以后才支持。

创建新节点的方法

- document.createElement("div")　　创建元素节点
- document.createAttribute("id")　　创建属性节点
- document.createTextNode("hello")　创建文本节点
- 一般将创建的新节点存在变量中，方便使用。

节点常用操作方法 1

- parentNode.appendChild(child)：将一个节点添加到指定父节点的子节点列表末尾。
- parentNode.replaceChild(newChild, oldChild)：用指定的节点替换当前节点的一个子节点，并返回被替换掉的节点。
- parentNode.insertBefore(newNode, referenceNode)：在参考节点之前插入一个拥有指定父节点的子节点，referenceNode 必须设置，如果 referenceElement 为 null 则 newNode 将被插入到子节点的末尾。
- parentNode.removeChild(child)：移除当前节点的一个子节点。这个子节点必须存在于当前节点中。

节点常用操作方法 2

- Node.cloneNode()：克隆一个节点，并且可以选择是否克隆这个节点下的所有内容。参数为 Boolean 布尔值，表示是否采用深度克隆,如果为 true,则该节点的所有后代节点也都会被克隆,如果为 false,则只克隆该节点本身，默认值为 true，节点下的内容会被克隆。
- 注意：克隆时，标签上的属性和属性值也会被复制，写在标签行内的绑定事件可以被复制，但是通过 JavaScript 动态绑定的事件不会被复制。

节点常用操作方法 3

- Node.hasChildNodes()：没有参数，返回一个 Boolean 布尔值，来表示该元素是否包含有子节点。
- Node.contains(child)：返回一个 Boolean 布尔值，来表示传入的节点是否为该节点的后代节点。

判断方法总结

- 有三种方法可以判断当前节点是否有子节点。
- node.firstChild !== null
- node.childNodes.length > 0
- node.hasChildNodes()

案例应用

- 动态创建列表
- 动态创建表格
- 选择水果

拉勾教育
—互联网人实战大学—

DOM 事件详解

掌握更多 DOM 事件操作方法

注册事件的其他方法1

- element.addEventListener() 方法。
- 参数：

  第一个参数：事件类型的字符串（直接书写”click”，不需要加 on）

  第二个参数：事件函数
- 同一个元素可以多次绑定事件监听，同一个事件类型可以注册多个事件函数
- 兼容性问题：不支持 IE9 以下的浏览器

注册事件的其他方法2

- element.attachEvent() 方法。
- 参数：

  第一个参数：事件类型的字符串（需要加 on）

  第二个参数：事件函数
- 同一个元素可以多次绑定事件监听，同一个事件类型可以注册多个事件函数
- 兼容性问题：只支持 IE10 及以下的浏览器

注册事件的兼容写法

- 自定义一个注册事件函数
- 参数：事件源，事件类型（不加 on），事件函数
- IE9 及以上的浏览器，使用 addEventListener 方法
- IE9 以下的浏览器，使用 attachEvent 方法
- 判断浏览器时，不需要判断它的版本，可以检测浏览器能力
- 浏览器能力检测：将某个方法的调用作为 if 语句的判断条件，如果浏览器认识该方法返回 true，否则返回 false。

移除事件的其他方法1

- element.removeEventListener() 方法。
- 参数：

  第一个参数：事件类型的字符串（直接书写”click”，不需要加 on）

  第二个参数：事件函数引用名
- 注意：没有办法移除一个匿名函数，所以在注册事件时需要单独声明一个有函数名的事件函数。
- 兼容性问题：不支持 IE9 以下的浏览器

移除事件的其他方法2

- element.detachEvent() 方法。
- 参数：

  第一个参数：事件类型的字符串（需要加 on）

  第二个参数：事件函数
- 注意：没有办法移除一个匿名函数，所以在注册事件时需要单独声明一个有函数名的事件函数。

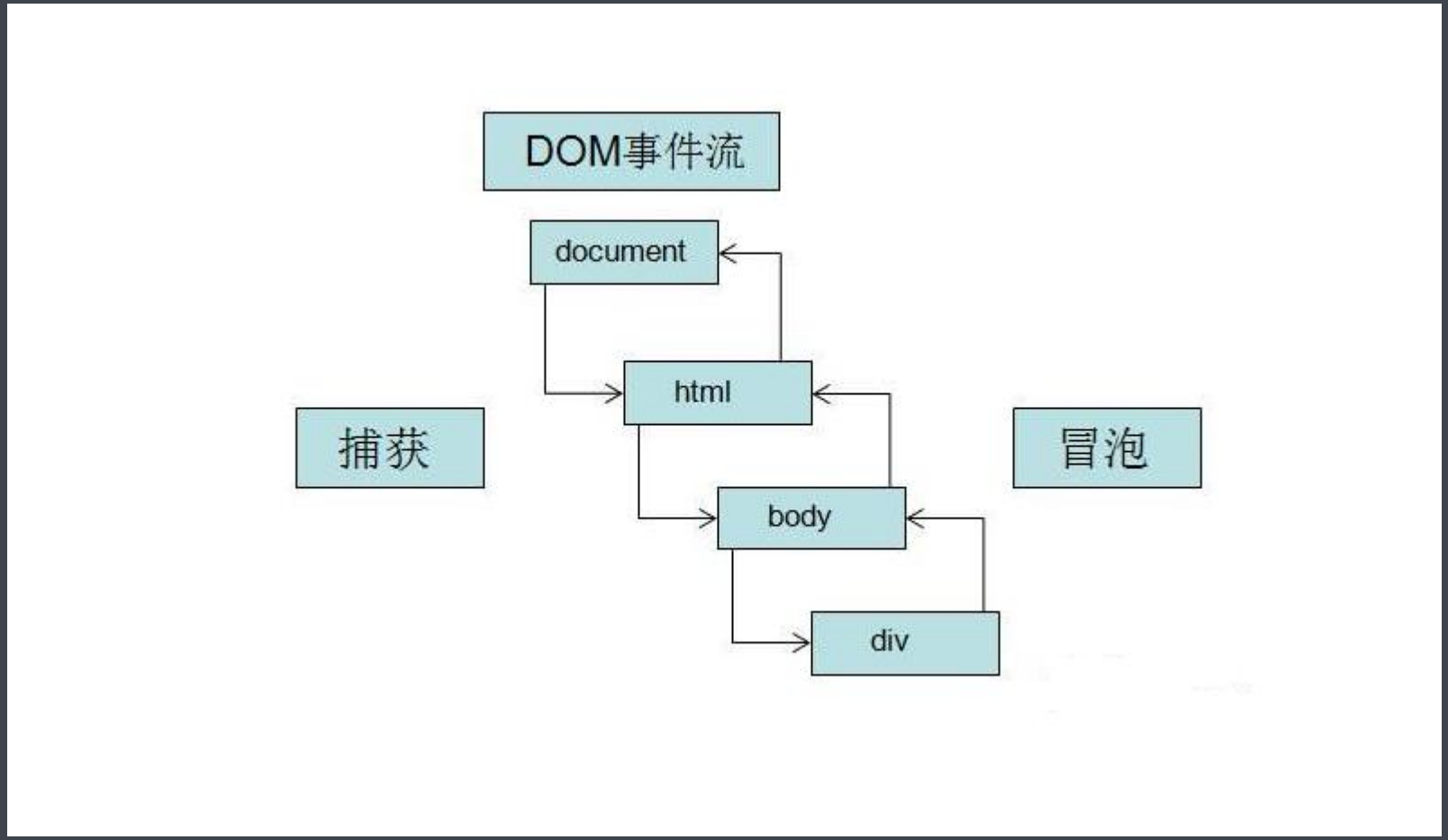
  兼容性问题：只支持 IE10 及以下的浏览器

移除事件的兼容写法

- 自定义一个移除事件函数
- 参数：事件源，事件类型（不加 on），事件函数
- IE9 及以上的浏览器，使用 removeEventListener 方法
- IE9 以下的浏览器，使用 detachEvent 方法

- 建议：将自己封装的一些常用函数和方法，放到一个单独的 .js 文件中。

DOM 事件流



事件流的三个阶段

- 第一个阶段：事件捕获
- 第二个阶段：事件执行过程
- 第三个阶段：事件冒泡
- addEventListener() 第三个参数为 false 时，事件冒泡
- addEventListener() 第三个参数为 true 时，事件捕获
- onclick 类型：只能进行事件冒泡过程，没有捕获阶段
- attachEvent() 方法：只能进行事件冒泡过程，没有捕获阶段

事件委托

- 利用事件冒泡，将子级的事件委托给父级加载
- 同时，需要利用事件函数的一个 e 参数，内部存储的是事件对象

事件对象

- 只要触发事件，就会有一个对象，内部存储了与事件相关的数据。
- e 在低版本浏览器中有兼容问题，低版本浏览器使用的是 window.event
- 事件对象常用的属性：

e.eventPhase	查看事件触发时所处的阶段
e.target	用于获取触发事件的元素
e.srcElement	用于获取触发事件的元素，低版本浏览器使用
e.currentTarget	用于获取绑定事件的事件源元素
e.type	获取事件类型
e.clientX/e.clientY	所有浏览器都支持，鼠标距离浏览器窗口左上角的距离
e.pageX/e.pageY	IE8 以前不支持，鼠标距离整个HTML页面左上顶点的距离

案例

- 图片跟随鼠标移动效果

取消默认行为和阻止事件传播的方式

- e.preventDefault()　　取消默认行为
- e.returnValue　　取消默认行为，低版本浏览器使用
- e.stopPropagation();　　阻止冒泡，标准方式
- e.cancelBubble = true;　　阻止冒泡，IE 低版本，标准中已废弃

其他事件类型

- MDN web 事件参考：https://developer.mozilla.org/zh-CN/docs/Web/Events

DOM 特效

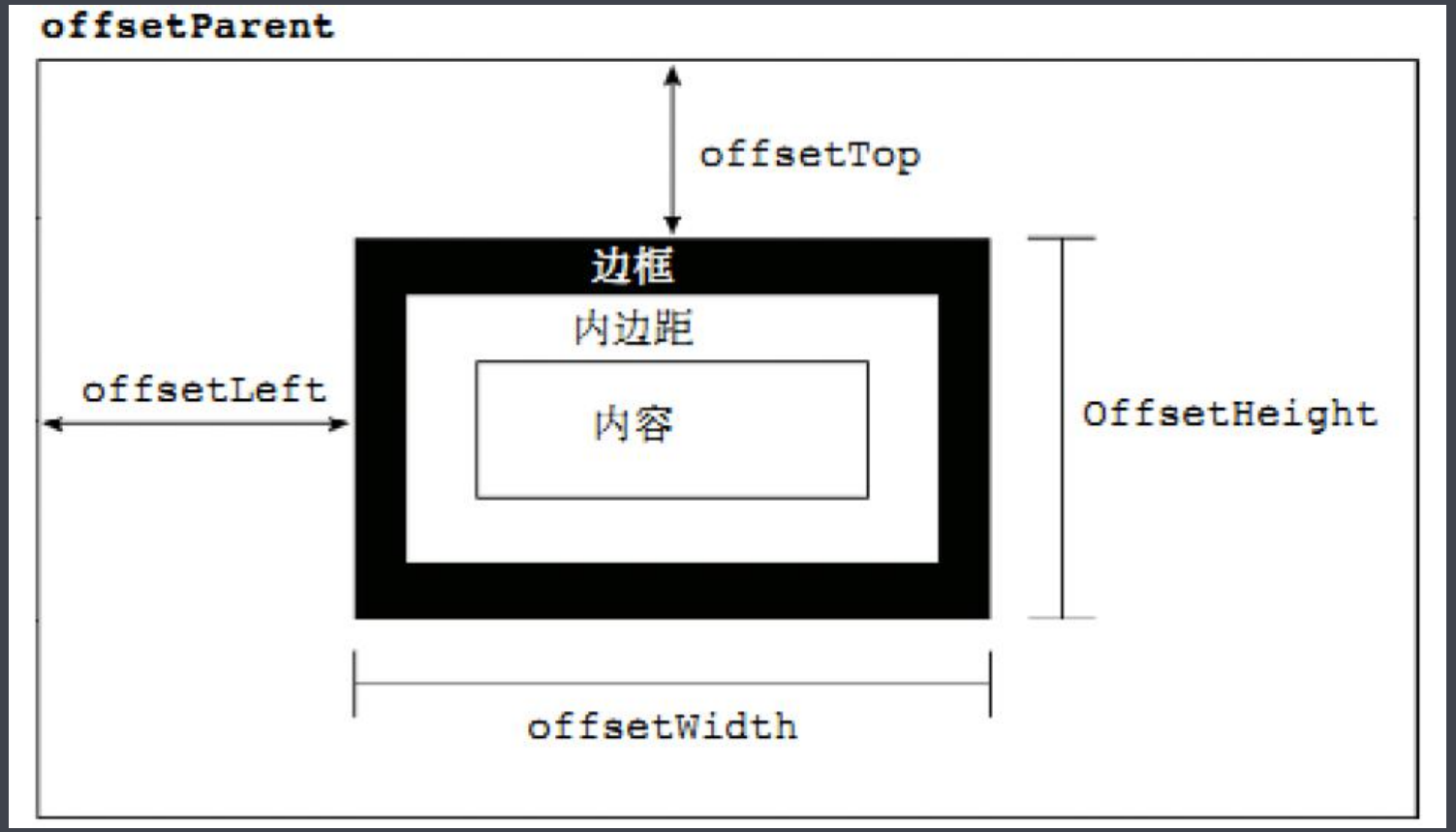
掌握 DOM 特效制作方法

DOM 提供了一套与元素自身有关的位置和大小的属性。

偏移量属性

- offsetParent 偏移参考父级，距离自己最近的有定位的父级，如果都没有定位参考body(html)
- offsetLeft/offsetTop 偏移位置
- offsetWidth/offsetHeight 偏移大小

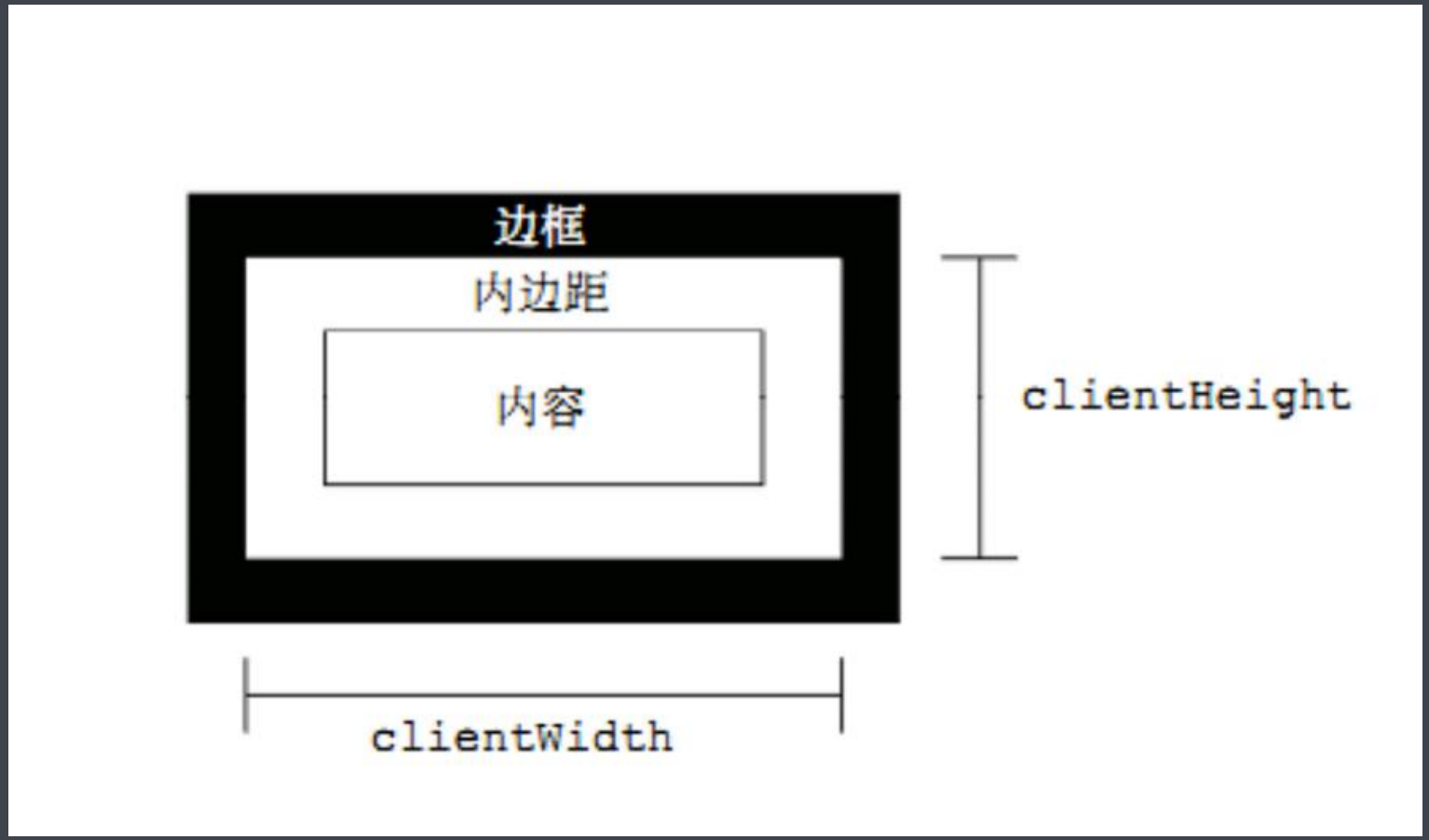
偏移量属性


offsetParent
offsetTop
边框
内边距
内容
offsetLeft
OffsetHeight
offsetWidth

客户端大小

- client 系列没有参考父级元素。
- clientLeft/clientTop　　边框区域尺寸，不常用
- clientWidth/clientHeight　　边框内部大小

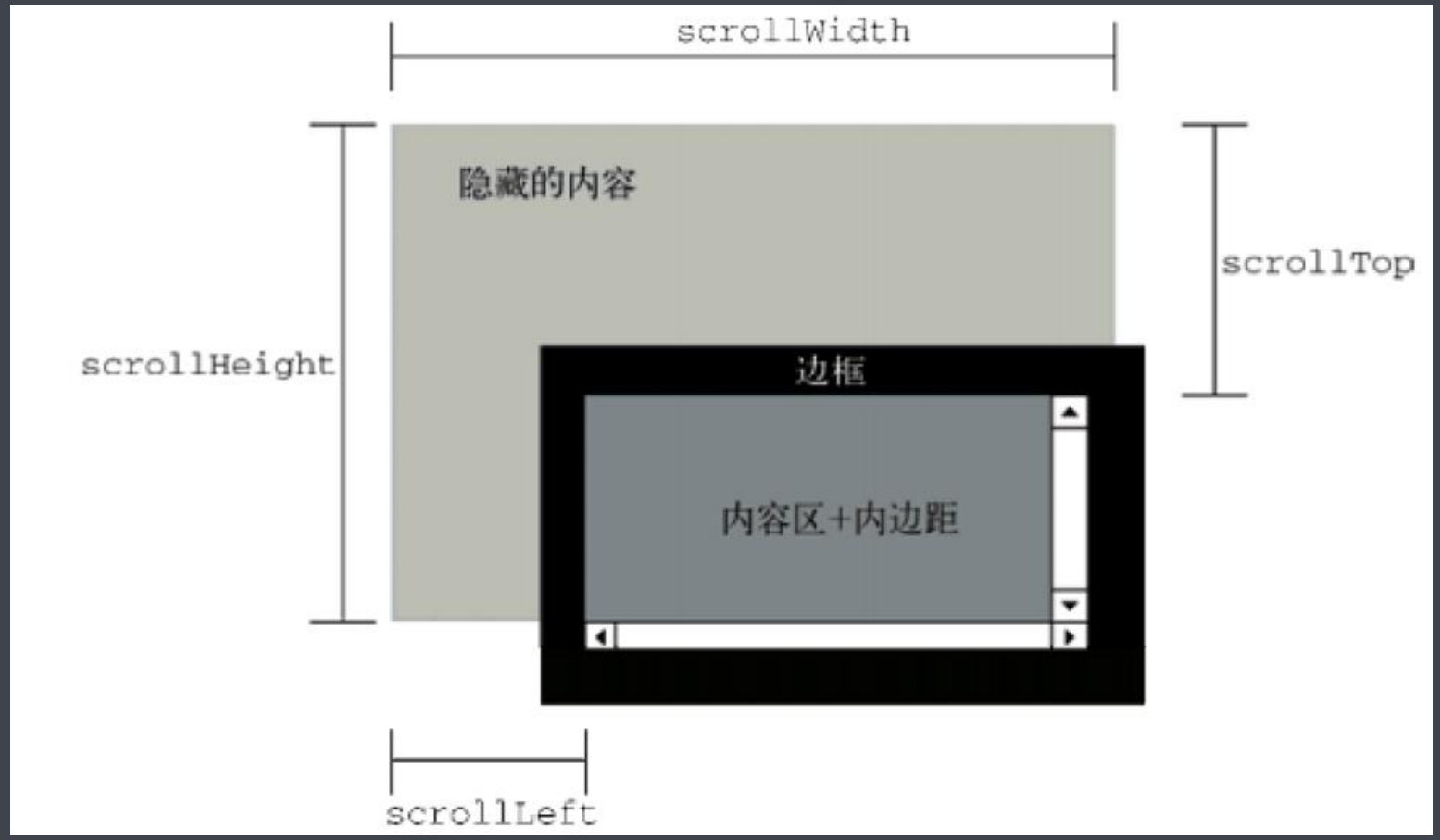
客户端大小



滚动偏移属性

- scrollLeft/scrollTop　　盒子内部滚动出去的尺寸
- scrollWidth/scrollHeight　　盒子内容的宽度和高度

滚动偏移属性


scrollWidth
隐藏的内容
scrollTop
scrollHeight
边框
内容区+内边距
scrollLeft

案例

- 拖拽案例
- 弹出登录窗口

拉勾教育

—互联网人实战大学—

下载「拉勾教育App」
获取更多内容