

DOM 概述

DOM

DOM 是 JavaScript 操作网页的接口，全称为“文档对象模型”（Document Object Model）。它的作用是将网页转为一个 JavaScript 对象，从而可以用脚本进行各种操作（比如增删内容）。

浏览器会根据 DOM 模型，将结构化文档（比如 HTML 和 XML）解析成一系列的节点，再由这些节点组成一个树状结构（DOM Tree）。所有的节点和最终的树状结构，都有规范的对外接口。

DOM 只是一个接口规范，可以用各种语言实现。所以严格地说，DOM 不是 JavaScript 语法的一部分，但是 DOM 操作是 JavaScript 最常见的任务，离开了 DOM，JavaScript 就无法控制网页。另一方面，JavaScript 也是最常用于 DOM 操作的语言。后面介绍的就是 JavaScript 对 DOM 标准的实现和用法。

节点

DOM 的最小组成单位叫做节点（node）。文档的树形结构（DOM 树），就是由各种不同类型的节点组成。每个节点可以看作是文档树的一片叶子。

节点的类型有七种。

- `Document`：整个文档树的顶层节点
- `DocumentType`：`doctype` 标签（比如 `<!DOCTYPE html>`）
- `Element`：网页的各种 HTML 标签（比如 `<body>`、`<a>` 等）
- `Attribute`：网页元素的属性（比如 `class="right"`）
- `Text`：标签之间或标签包含的文本
- `Comment`：注释
- `DocumentFragment`：文档的片段

浏览器提供一个原生的节点对象 `Node`，上面这七种节点都继承了 `Node`，因此具有一些共同的属性和方法。

节点树

一个文档的所有节点，按照所在的层级，可以抽象成一种树状结构。这种树状结构就是 DOM 树。它有一个顶层节点，下一层都是顶层节点的子节点，然后子节点又有自己的子节点，就这样层层衍生出一个金字塔结构，倒过来就像一棵树。

浏览器原生提供 `document` 节点，代表整个文档。

```
```javascript
document
// 整个文档树
```

...

文档的第一层有两个节点，第一个是文档类型节点（`<!doctype html>`），第二个是 HTML 网页的顶层容器标签`<html>`。后者构成了树结构的根节点（root node），其他 HTML 标签节点都是它的下级节点。

除了根节点，其他节点都有三种层级关系。

- 父节点关系（parentNode）：直接的那个上级节点
- 子节点关系（childNodes）：直接的下级节点
- 同级节点关系（sibling）：拥有同一个父节点的节点

DOM 提供操作接口，用来获取这三种关系的节点。比如，子节点接口包括`firstChild`（第一个子节点）和`lastChild`（最后一个子节点）等属性，同级节点接口包括`nextSibling`（紧邻在后的那个同级节点）和`previousSibling`（紧邻在前的那个同级节点）属性。