

# DOM 概述

DOM 是 JavaScript 操作网页的接口，全称为“文档对象模型”（Document Object Model）。

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。

## 1. 节点

DOM 的最小组成单位叫做节点（node），文档的树形结构（DOM 树），就是由不同类型的节点树组成。

节点的类型有 7 种：

- **Document**：文档节点，整个文档树的顶层节点
- **DocumentType**：文档类型节点，**doctype** 标签（HTML5 的文档标签是 `<!DOCTYPE html>`）
- **Element**：元素节点，网页的各种 HTML 标签（比如 `<body>` `<title>`）
- **Attr**：属性节点，网页元素的属性（比如 `charset="UTF-8"`）
- **Text**：文本节点，标签包含的文本
- **Comment**：注释节点，注释
- **DocumentFragment**：文档片段节点，文档的片段，文档片段可插入 DOM 树中

## 2. 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，又像一棵树。

浏览器原生提供 **document** 节点，代表整个文档。

```
document;  
// #document    整个节点树
```

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。

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body></body>
</html>
```

以上面 DOM 结构为例：

```
document.firstChild;           // <!DOCTYPE html>
document.lastChild.previousSibling; // <!DOCTYPE html>
document.lastChild.firstChild.lastElementChild;
// lastElementChild 表示最后一个子元素节点 <title>Document</title>

document.lastChild.firstChild.lastElementChild ===
document.lastChild.firstChild.lastChild;
// true, 因为 <title>Document</title> 即是它所在父节点的最后一个子节点，同时也是元素节点
```