## DOM 节点操作

- 获取 DOM 节点
- attribute
- property

慕课网

## 获取 DOM 节点

```
const div1 = document.getElementById('div1') // 元素
const divList = document.getElementsByTagName('div')  // 集合
console.log(divList.length)
console.log(divList[0])

const containerList = document.getElementsByClassName('.container')
 // 集合
const pList = document.querySelectorAll('p') // 集合
```

慕课网

## DOM 节点的 property

```
const pList = document.querySelectorAll('p')
const p = pList[0]
console.log(p.style.width)  // 获取样式
p.style.width = '100px'  // 修改样式
console.log(p.className)  // 获取 class
p.className = 'p1'  // 修改 class

// 获取 nodeName 和 nodeType
console.log(p.nodeName)
console.log(p.nodeType)
```

慕课网

## property 和 attribute

- property：修改对象属性，不会体现到 html 结构中
- attribute：修改 html 属性，会改变 html 结构
- 两者都有可能引起 DOM 重新渲染

慕课网

# DOM 结构操作

- 新增/插入节点
- 获取子元素列表，获取父元素
- 删除子元素

慕课网

## 新增/插入节点

```
const div1 = document.getElementById('div1')
// 添加新节点
const p1 = document.createElement('p')
p1.innerHTML = 'this is p1'
div1.appendChild(p1) // 添加新创建的元素
// 移动已有节点。注意是移动！！！
const p2 = document.getElementById('p2')
div1.appendChild(p2)
```

慕课网

## 删除节点

```
const div1 = document.getElementById('div1')
const child = div1.childNodes
div1.removeChild(child[0])
```

慕课网

## 获取子元素列表 & 获取父元素

```
// 获取子元素列表
const div1 = document.getElementById('div1')
const child = div1.childNodes

// 获取父元素
const div1 = document.getElementById('div1')
const parent = div1.parentNode
```

慕课网

# DOM 性能

- DOM 操作非常“昂贵”，避免频繁的 DOM 操作
- 对 DOM 查询做缓存
- 将频繁操作改为一次性操作

慕课网

# DOM 查询做缓存

```
// 不缓存 DOM 查询结果
for (let i = 0; i < document.getElementsByTagName('p').length; i++) {
    // 每次循环，都会计算 length ，频繁进行 DOM 查询
}

// 缓存 DOM 查询结果
const pList = document.getElementsByTagName('p')
const length = pList.length
for (let i = 0; i < length; i++) {
    // 缓存 length ，只进行一次 DOM 查询
}
```

慕课网

# 将频繁操作改为一次性操作

```
const listNode = document.getElementById('list')

// 创建一个文档片段，此时还没有插入到 DOM 树中
const frag = document.createDocumentFragment()

// 执行插入
for(let x = 0; x < 10; x++) {
    const li = document.createElement("li")
    li.innerHTML = "List item " + x
    frag.appendChild(li)
}

// 都完成之后，再插入到 DOM 树中
listNode.appendChild(frag)
```

慕课网