



06. 객체인가? 구조인가? – DOM을 둘러싼 두 가지 시선

1. DOM은 구조인가, 객체인가?

지금까지 우리는 DOM이 HTML에서 어떻게 만들어지는지, 그리고 자바스크립트를 통해 DOM에 어떻게 접근할 수 있는지를 하나하나 배워왔습니다. 그런데 많은 분들이 이 시점에서 하나의 중요한 질문에 부딪히곤 합니다. “DOM은 도대체 무엇인가요? 객체인가요? 구조인가요?” 이 질문은 단순한 개념 정리에 그치지 않고, 앞으로 DOM을 자유자재로 다루기 위한 **사고방식의 전환점**이 됩니다.

→ DOM을 트리 구조로 설명하는 경우도 있고, 반대로 `document.body` 처럼 객체의 속성에 접근하듯 설명하기도 합니다.

→ 그렇다면 어떤 관점이 맞는 걸까요?

결론부터 말하자면, **DOM은 구조이면서 동시에 객체들의 집합**입니다.

HTML 문서는 **구조화된 객체 트리**로 변환되며, 그 각 요소는 자바스크립트로 다룰 수 있는 독립적인 객체가 됩니다.

2. 대학교 조직도 비유: DOM은 구조이자 객체

이해를 돕기 위해 실생활에 비유해보겠습니다. 여러분은 대학교 조직도를 본 적이 있나요? 예를 들어 “서울대학교”라는 대학이 있다고 가정해봅시다. 그 안에는 여러 단과대학이 있고, 단과대학 안에는 학과가 있으며, 각 학과에는 교수와 학생들이 소속되어 있습니다.

→ 이 전체는 명확한 **계층 구조**를 가지고 있습니다.

→ 그러나 동시에 각 구성원은 이름, 나이, 직책, 연락처 등 다양한 속성을 가진 **개별 객체**이기도 하죠.

DOM도 이와 동일합니다.

문서 전체는 **하나의 트리 구조**로 연결되어 있고, 각 요소는 **객체로서 속성과 기능을 가진 독립적인 단위**입니다.

3. DOM 트리의 구조를 시각화해보기

아래와 같은 간단한 HTML을 살펴봅시다:

```
<div>
  <h1>제목</h1>
  <p>설명입니다</p>
</div>
```

이 HTML은 브라우저가 해석한 뒤 다음과 같은 트리 구조로 변환됩니다:

```
Document
├── div
│   ├── h1
│   │   └── "제목"
│   └── p
│       └── "설명입니다"
```

- 각 태그는 부모-자식 관계를 가진 노드로 연결되며, 이 전체가 DOM 트리입니다.
- `div` 는 두 개의 자식을 가지며, `h1` 과 `p` 는 각각 텍스트 노드를 자식으로 가지고 있습니다.
- 이 구조는 마치 **가계도**처럼 위계적이며, HTML의 중첩 구조를 그대로 반영합니다.

4. DOM 구조에 객체처럼 접근하기

그렇다면 이 DOM 구조를 자바스크립트에서는 어떻게 접근할 수 있을까요? 다음 코드를 보겠습니다:

```
const div = document.querySelector("div");
console.log(div.nodeName); // "DIV"
console.log(div.children.length); // 2
console.log(div.children[0].textContent); // "제목"
```

- `querySelector("div")` 로 첫 번째 `div` 요소를 선택하고,
 - 그 자식 요소들에 `children` 속성을 통해 접근하고,
 - `textContent` 로 텍스트 값을 꺼냅니다.
- 이 모든 과정은 **객체의 속성 탐색과 동일한 방식**으로 이루어집니다.

→ DOM 트리의 각 요소는 실제로 **자바스크립트 객체**입니다.

5. console.dir(document)로 구조를 시각화하기

개발자 도구에서 다음 명령어를 실행해보세요:

```
console.dir(document);
```

→ `console.log(document)` 는 실제 HTML 구조를 보여주지만,

→ `console.dir(document)` 는 **객체 내부 속성 트리**를 시각화해서 보여줍니다.

→ 즉, 브라우저 내부에서 DOM은 **구조이자 객체의 집합**이라는 점을 이 명령어를 통해 명확히 확인할 수 있습니다.

6. 중첩된 구조 탐색: 계층 + 접근

아래의 HTML 문서를 보겠습니다:

```
<body>
  <section>
    <article>
      <h2>제목</h2>
    </article>
  </section>
</body>
```

이 문서는 다음과 같은 트리 구조를 가집니다:

```
body
├── section
│   └── article
│       └── h2
│           └── "제목"
```

→ 브라우저는 이 구조를 메모리에 그대로 저장합니다.

→ 그래서 다음과 같은 접근이 가능해지는 것입니다:

```
document.body.children[0].children[0].children[0].textContent;
```

→ 이처럼 DOM 트리는 계층 구조를 기반으로 탐색할 수 있으며,

→ 각 요소는 객체로서 속성과 메서드를 갖는 노드입니다.

7. DOM의 이중성: 구조 + 객체

DOM을 완전히 이해하기 위해서는 이 두 가지 개념을 함께 받아들여야 합니다:

- **구조적 측면:** DOM은 부모-자식-형제 노드로 구성된 계층 트리입니다.
- **객체적 측면:** DOM의 각 노드는 자바스크립트 객체이며, 속성과 기능을 가집니다.

→ `nodeType`, `nodeName`, `childNodes`, `parentNode` 등은 DOM 노드가 가진 대표적인 속성들입니다.

→ 구조로 접근하되 객체로 다루고, 객체로 다루되 구조를 인식할 줄 알아야 진짜 DOM을 자유자재로 활용할 수 있게 됩니다.

✓ 핵심 요약

정리하자면, DOM은 **HTML 문서를 표현하는 구조화된 트리**이며,

이 트리는 **객체들로 구성되어 있고**,

자바스크립트로 이 객체에 접근하고 조작할 수 있습니다.

- 구조만 알면 탐색은 가능하지만 조작은 어렵고,
- 객체로만 이해하면 전체 흐름을 놓치기 쉽습니다.

→ **DOM = 구조 + 동작**

→ **DOM = 트리 + 객체**

이 두 관점을 함께 이해하는 것이 DOM 마스터리의 출발점입니다.