📘 DOM 완전 정복 – 01. HTML만으로는 부족했던 이유 – 구조만 있고, 조작은 없던 시절

1. HTML의 본질과 구조적 한계

HTML(HyperText Markup Language)은 '하이퍼텍스트 문서를 표현하기 위한 마크업 언어'입니다. 다시 말해, HTML은 본래 프로그래밍 언어가 아니며, 복잡한 논리나 제어 흐름을 처리할 수 없습니다. HTML의 목적은 오직 한 가지, **정보의 구조를 명시적으로 표현**하는 것입니다.

예를 들어 다음과 같은 코드는 하나의 정적인 문서를 구성합니다:

```
<h1>안녕하세요</h1>
<p>이것은 문단입니다.</p>
```

이 구조는 사용자에게 내용을 보여주기는 하지만, 사용자의 어떤 행동에도 반응하지 않습니다. 마치 종이에 인쇄된 책처럼 읽는 기능만 있는 것입니다. 이처럼 HTML만으로는 **버튼 클릭, 입력 반응, 동적 전환 등 어떤 상호작용도 구현할 수 없습니다.**

2. 상호작용의 요구와 자바스크립트의 등장

현대 웹은 단순히 정보를 전달하는 문서가 아니라, 사용자의 행동에 반응하고 실시간으로 변화하는 **소프트웨어 플랫폼**입니다. 예를 들어,

- 버튼을 누르면 레이어 팝업이 뜨고
- 댓글을 달면 바로 화면에 반영되며
- 마우스를 올리면 애니메이션이 재생되고
- 로그인 상태에 따라 메뉴가 바뀝니다

이러한 기능을 가능하게 해주는 것이 바로 **자바스크립트**입니다. 자바스크립트는 동적인 동작을 정의하고, 사용자와의 상호작용을 코드로 제어할 수 있게 해줍니다. 하지만 여기서 중요한 질문

이 생깁니다:

> 자바스크립트는 HTML을 직접 수정할 수 있는가?

정답은 “아니오”입니다.

---

3. DOM의 등장 – 문서를 객체로 바꾼 모델

자바스크립트는 HTML 파일을 직접 수정하지 않습니다. 대신, **브라우저가 HTML을 해석해 메모리 상에 구성한 객체 구조**, 즉 DOM(Document Object Model)을 조작합니다.

> DOM이란?
>
> - Document: 문서
> - Object: 객체
> - Model: 구조화된 표현

DOM은 브라우저가 HTML을 파싱하여 구성한 **트리 형태의 객체 집합**입니다. 자바스크립트는 이 DOM 객체를 읽고 수정함으로써 화면의 내용을 동적으로 바꾸는 것입니다.

예를 들어, 다음과 같은 HTML이 있다고 가정합시다:

```
<body>
  <div>
    <p>Hello</p>
  </div>
</body>
```

브라우저는 이를 다음과 같은 트리 구조로 내부 메모리에 구성합니다:

```
Document
 └── body
    └── div
       └── p
          └── "Hello"
```

4. 트리 구조란 무엇인가?

HTML은 단순히 들여쓰기된 문장이 아니라, 논리적으로 **부모와 자식 요소가 중첩(nesting)된 계층 구조**입니다. 이를 컴퓨터가 이해할 수 있도록 트리(tree) 형태의 데이터로 바꾸어 저장하는 것입니다.

📌 현실 세계 예시: 조직도

```
회사
 └── 부서
    └── 팀
       └── 팀원
```

📌 DOM 구조 예시

```
Document
 └── body
    └── div
       └── p
          └── "Hello"
```

이 구조는 실제로 메모리에 다음과 같은 객체 형태로 존재합니다:

```
{
  body: {
    div: {
      p: {
        text: "Hello"
      }
    }
  }
}
```

5. DOM 조작 예제

```
<button id="btn">눌러보세요</button>
<p id="text">안녕하세요</p>

<script>
  const button = document.getElementById("btn");
  const text = document.getElementById("text");

  button.addEventListener("click", () => {
    text.textContent = "DOM을 통해 바뀌었어요!";
  });
</script>
```

이 코드에서는 다음과 같은 일이 일어납니다:

1. 브라우저는 HTML을 읽고 DOM 트리를 생성합니다.
2. 자바스크립트가 `document.getElementById()`로 해당 DOM 요소를 가져옵니다.
3. 버튼에 클릭 이벤트 리스너를 추가합니다.
4. 클릭 시, 텍스트 DOM 노드의 `.textContent`를 변경합니다.
5. 화면의 내용이 즉시 바뀝니다.

⚠️ 이 모든 과정은 **HTML 파일을 수정하는 것이 아니라**, 브라우저 메모리 상의 **DOM 객체를 실시간으로 조작**한 결과입니다.

---

6. DOM이 없었다면?

- 자바스크립트는 HTML의 어떤 요소도 찾을 수 없습니다.
- 클릭, 입력, 스크롤 등 사용자 동작에 반응할 수 없습니다.
- 웹페이지는 언제나 정적인 상태로만 존재하게 됩니다.

---

7. DOM의 핵심 요약

구분	설명
HTML	구조를 정의한 설계도

구분	설명
DOM	구조를 객체화한 메모리 트리
JS	객체 트리를 조작하여 웹을 움직이게 만드는 제어자

[HTML] → [DOM] → [JavaScript]
설계도 실제 구조물 기능을 수행하는 시스템

---

8. 시각적 구조 요약

HTML 문서
├── 구조만 존재 (읽기 전용)
├── 클릭, 입력 등 반응 없음
└── 정적인 텍스트 기반

브라우저 → DOM 생성
├── 객체로 구조화된 메모리 트리
├── 각 요소는 노드(Node)
└── 부모-자식 계층으로 연결

자바스크립트
├── DOM 객체 탐색 (document.getElementById 등)
├── 속성 수정 (textContent, style 등)
└── 이벤트 처리 (addEventListener 등)

---

9. 핵심 개념 정리 퀴즈

- Q1. 자바스크립트는 HTML 파일을 직접 수정할 수 있는가?

  → ❌ 아니다. DOM 객체를 조작한다.

- Q2. DOM은 왜 필요한가?

  → HTML을 자바스크립트가 다룰 수 있게 객체로 바꿔주는 브라우저의 메모리 구조이기 때문이다.

- Q3. DOM 트리는 무엇인가?

  → HTML 요소들을 객체로 변환한 후, 부모-자식 관계로 연결한 계층 구조이다.

✅ 마무리 요약

HTML은 '구조를 정의한 문서'에 불과합니다. 이 정적인 문서를 **자바스크립트가 조작할 수 있도록 객체화**한 것이 DOM이며, 이 DOM 덕분에 웹은 동적인 반응을 구현할 수 있게 되었습니다. DOM을 이해한다는 것은 곧 **현대 웹 개발의 기본 언어를 습득하는 것**과 같습니다.
