

12. `getElementsByTagName`으로 태그명으로 선택하기

1. 클래스 다음은 태그다! DOM 요소를 태그로 한 번에 선택하는 방법

이제 우리는 DOM 트리 속에서 여러 요소를 한꺼번에 선택하는 방법들을 하나씩 배워가고 있습니다.

지난 강의에서는 `getElementsByClassName()` 을 통해 공통된 클래스 이름을 가진 요소들을 한꺼번에 선택하는 방법을 배웠습니다.

이번 강의에서는 요소의 **태그 이름 자체**를 기준으로 여러 요소를 선택하는 방법인 `getElementsByTagName()` 을 배워보겠습니다.

→ 이 메서드는 구조는 단순하지만, 실제로 매우 강력하고 유용한 기능을 제공합니다.

2. 이름 그대로 태그 이름으로 선택한다

`getElementsByTagName()` 은 말 그대로 특정 태그 이름을 가진 모든 요소를 한꺼번에 가져오는 메서드입니다.

- **get** → 가져온다
- **Elements** → 여러 개의 요소를
- **ByTagName** → 태그 이름 기준으로

→ 예: `<p>`, `<div>`, ``, ``, `<input>` 등

3. 기본 문법은 이렇게 사용한다

다음 코드는 HTML 문서 내의 모든 `<p>` 태그를 선택하는 예시입니다:

```
const paragraphs = document.getElementsByTagName("p");
```

→ `paragraphs` 에는 `<p>` 요소들이 **HTMLCollection** 형태로 저장됩니다.

4. HTMLCollection은 배열처럼 생겼지만 배열이 아니다

`getElementsByName()`의 반환값은 **HTMLCollection**이라는 유사 배열 객체입니다.

특징	설명
배열처럼 인덱스로 접근 가능	<code>paragraphs[0]</code> , <code>paragraphs[1]</code>
<code>.length</code> 로 개수 확인 가능	<code>paragraphs.length</code>
<code>forEach</code> , <code>map</code> 사용 불가	배열 메서드는 동작하지 않음

→ 반복문은 보통 `for` 또는 `for...of`를 사용합니다.

5. 실전 예제: 모든 `<p>` 요소에 스타일 적용하기

다음 HTML 문서를 예로 들어봅시다:

```
<p>첫 번째 문단입니다.</p>
<p>두 번째 문단입니다.</p>
<p>세 번째 문단입니다.</p>
```

이 HTML에서 `<p>` 요소는 3개입니다. 자바스크립트로 한꺼번에 선택하고 색상을 바꾸려면 다음과 같이 작성합니다:

```
const paragraphs = document.getElementsByTagName("p");

for (let i = 0; i < paragraphs.length; i++) {
  paragraphs[i].style.color = "blue";
}
```

→ 실행 결과: 세 문단 모두 **파란색 글자**로 표시됩니다.

6. ``, `<td>`, `<input>` 에도 그대로 적용 가능

`getElementsByName()`은 `<p>` 뿐 아니라 다음과 같은 다양한 태그에도 동일하게 적용됩니다:

- 목록 항목 → ``
- 테이블 셀 → `<td>`

- 폼 입력 → `<input>`

→ 태그 이름이 같기만 하면 한 번에 모두 선택하여 조작할 수 있습니다.

7. 특정 요소 내부에서만 검색할 수도 있다

중요한 기능 중 하나는, 특정 요소 안에서만 검색을 제한할 수 있다는 점입니다.

예를 들어, `id="container"` 인 박스 안에 있는 `` 들만 찾고 싶다면 이렇게 작성합니다:

```
const box = document.getElementById("container");
const spans = box.getElementsByTagName("span");
```

→ `spans` 에는 `#container` 내부에 있는 `` 요소들만 담깁니다.

→ 문서 전체가 아닌 부분 범위 탐색이 가능하다는 점이 매우 실용적입니다.

8. 콘솔로 실제 확인해보기

선택된 요소들을 눈으로 직접 확인하고 싶다면 다음을 사용합니다:

```
console.dir(spans);
```

→ `console.dir()` 은 해당 요소의 속성 구조를 트리 형태로 시각화해주는 디버깅 도구입니다.

→ `childNodes`, `attributes`, `innerText` 등 모든 정보를 확인할 수 있습니다.

✓ 핵심 요약

- `getElementsByTagName("태그이름")` 은 해당 태그를 가진 모든 요소를 한 번에 선택합니다.
- 반환값은 **HTMLCollection**이며, 인덱스 접근 및 `.length` 사용 가능
- 배열 메서드는 사용 불가 (`for` 루프 사용)
- 전체 문서뿐 아니라 특정 요소 내부에서 부분 탐색도 가능
- 실무에서도 ``, `<input>`, `<div>` 등 자주 활용되는 선택자

→ 태그 이름만 알고 있으면 범용적으로 많은 요소를 쉽게 조작할 수 있는 필수 도구입니다.

다음 강의 예고: `querySelector()` 와 `querySelectorAll()` 로 CSS처럼 선택하기

다음 시간에는 훨씬 더 강력하고 유연한 DOM 선택자인

**** `querySelector()` 와 `querySelectorAll()` ****을 배워보겠습니다.

→ 이 메서드들은 CSS 선택자 문법을 그대로 사용할 수 있기 때문에,

→ 더 복잡하고 세밀한 조건의 요소 선택이 가능하며,

→ 실무에서 가장 널리 쓰이는 선택 방법입니다.

본격적인 **DOM 마스터리의 실전 단계**가 시작됩니다.

그럼 다음 강의에서 뵙겠습니다! 감사합니다! 🙌