



## 3강 - 속성 노드

어트리뷰트(attribute)

HTML 속성 조작하기

속성 추가, 변경하기

속성값 참조하기

속성 삭제하기

data 어트리뷰트와 dataset 프로퍼티

스타일 조작

인라인 스타일 조작

클래스 조작

### 어트리뷰트(attribute)

```
<input id="name" type="text" value="Soongu">
```

- HTML 요소는 여러 개의 속성(어트리뷰트)을 가질 수 있습니다.
- 문서가 파싱될 때 HTML 요소의 속성은 어트리뷰트 노드로 변환되어 요소 노드의 형제 노드로 추가됩니다.
- 이 때 속성 하나당 하나의 어트리뷰트 노드가 생성됩니다. 즉 위 예시의 경우 3개의 어트리뷰트 노드가 생성됩니다.
- 모든 어트리뷰트 노드의 참조는 유사 배열 객체인 NamedNodeMap 객체에 담겨서 요소 노드의 `attributes` 프로퍼티에 저장됩니다.

## HTML 속성 조작하기

### 속성 추가, 변경하기

#### `$elementNode.setAttribute(name, value)`

- `setAttribute()` 메서드는 요소노드의 속성값을 추가하거나 변경할 때 사용합니다.
- 첫번째 인자 `name`에 속성이름 (ex: `src`)을 넣고 두번째 인자 `value`에 추가하거나 변경할 속성값을 전달합니다.
- 속성이름이 요소노드에 이미 존재한다면 수정되고, 존재하지 않는다면 추가됩니다.

```
<input id="mail-input" type="email" name="mail" value="abc1234@naver.com">
```

```
const $input = document.getElementById('mail-input');

$input.setAttribute('type', 'text'); // type속성의 값이 text로 변경됨
$input.setAttribute('placeholder', '메일을 작성하세요'); // placeholder속성이 추가됨
```

## 속성값 참조하기

### **\$elementNode.getAttribute(name)**

- getAttribute() 메서드는 속성에 저장되어 있는 속성값을 가져옵니다.

```
<input id="mail-input" type="email" name="mail" value="abc1234@naver.com">

const $input = document.getElementById('mail-input');

const email = $input.getAttribute('value');

console.log(email); // abc1234@naver.com
```

## 속성 삭제하기

### **\$elementNode.removeAttribute(name)**

- removeAttribute() 메서드는 요소에서 해당 속성을 제거합니다.

## data 어트리뷰트와 dataset 프로퍼티

```
<ul class="users">
  <li id="1" data-user-id="8841" data-role="common">Kim</li>
  <li id="2" data-user-id="4967" data-role="admin">Hong</li>
</ul>
```

- data 어트리뷰트와 dataset 프로퍼티를 사용하면 HTML요소와 자바스크립트 간에 데이터를 교환할 수 있습니다.
- data 어트리뷰트는 data-role과 같이 data- 접두사를 사용하여 임의의 이름을 붙여 사용합니다.
- 이렇게 지정한 data- 속성은 자바스크립트에서 \$elementNode.dataset 프로퍼티로 조회할 수 있습니다.
- 자바스크립트는 케밥케이스로 되어있는 data- 속성을 카멜케이스로 변환하여 가지고 있습니다.  
ex) data-user-id -> userId

```
const $users = document.querySelector('.users');

const [$u1, $u2] = [...$users.children];

console.log($u1.dataset.userId); // 8841
console.log($u2.dataset.role);   // admin
```

## 스타일 조작

### 인라인 스타일 조작

**\$elementNode.style**

- style 프로퍼티는 요소 노드의 인라인 스타일을 취득하거나 추가 또는 변경합니다.
- style 프로퍼티를 참조하면 CSSStyleDeclaration 타입의 객체를 반환하는데 해당 객체는 css 프로퍼티에 대응하는 프로퍼티를 가지고 있습니다.
- css속성을 사용할 때는 카멜케이스를 적용합니다.

### 클래스 조작

**\$elementNode.className**  
**\$elementNode.classList**

- className 프로퍼티는 HTML 요소의 class 속성 값을 취득하거나 변경합니다.

- classList 프로퍼티는 class 속성의 정보를 담은 DOMTokenList 객체를 반환합니다.
- 일반적으로 className으로 클래스를 조작하기보다는 좀 더 유용한 메서드를 지원하는 classList로 클래스를 제어하는 것이 좋습니다.

## classList가 제공하는 유용한 메서드

메서드명	설명
add(...className)	인수로 전달한 클래스를 추가
remove(...className)	인수로 전달한 클래스를 제거
item(index)	index번째 해당하는 클래스를 반환
contains(className)	인수로 전달한 클래스명과 일치하는 클래스 속성이 있는지 확인
replace(old, new)	첫번째 인수로 전달한 클래스를 두번째 인수로 전달한 클래스로 교체
toggle(className)	인수로 전달한 클래스명이 속성으로 이미 존재하면 삭제하고 존재하지 않으면 추가함