

삼육대학교 SW융합교육원 김진호

자바스크립트 기초교육

1-2. DOM (노드 탐색)

1-2-1. child node

1-2-2. parent node

1-2-3. sibling node

❖ 1-2.DOM

1-2. DOM (노드 탐색)

요소를 취득한 다음, 취득한 요소를 기점으로 DOM 트리의 노드를 옮겨다니며 부모, 형제, 자식 노드 등을 탐색(traversinig)해야 할 때가 있다. DOM 트리 상의 노드를 탐색할 수 있도록 Node, Element 인터페이스는 트리 탐색 프로퍼티를 제공한다. 노드 탐색 프로퍼티는 모두 참조만 가능한 읽기 전용 접근자 프로퍼티이다. 읽기 전용 접근자 프로퍼티에 값을 할당하면 아무런 에러 없이 무시된다.

■ 공백 텍스트 노드

- HTML 요소 사이의 스페이스, 탭, 줄바꿈(개행) 등의 공백 문자는 텍스트 노드를 생성한다.

❖ 1-2.DOM

1-2. DOM (노드 탐색)

1-2-1. child node (자식 노드)

- **Node.prototype.childNodes** : 자식 노드(요소 노드, 텍스트 노드)를 탐색하여 NodeList에 담아 반환
- **Node.prototype.firstChild** : 첫 번째 자식 노드(요소 노드, 텍스트 노드) 반환
- **Node.prototype.lastChild** : 마지막 자식 노드(요소 노드, 텍스트 노드) 반환

```
<ol id="node">
  <li>Node.prototype.childNodes : 자식 노드(요소 노드, 텍스트 노드)를 탐색하여
  NodeList에 담아 반환</li>
  <li>Node.prototype.firstChild : 첫 번째 자식 노드(요소 노드, 텍스트 노드) 반환</li>
  <li>Node.prototype.lastChild : 마지막 자식 노드(요소 노드, 텍스트 노드) 반환</li>
</ol>
```

❖ 1-2.DOM

1-2. DOM (노드 탐색)

1-2-1. child node (자식 노드)

```
const $node = document.getElementById('node');  
  
// NodeList 반환, 요소 노드뿐만 아니라 텍스트 노드도 포함  
console.log($node.childNodes);  
// 첫 번째 자식 노드 탐색  
console.log($node.firstChild);  
// 마지막 자식 노드 탐색  
console.log($node.lastChild);
```

❖ 1-2.DOM

1-2. DOM (노드 탐색)

1-2-1. child node (자식 노드)

- **Element.prototype.children** : 자식 노드 중 요소 노드만 탐색하여 HTMLCollection에 담아 반환
- **Element.prototype.firstElementChild** : 첫 번째 자식 요소 노드 반환
- **Element.prototype.lastElementChild** : 마지막 자식 요소 노드 반환

```
<ol id="element">
  <li>Element.prototype.children : 자식 노드 중 요소 노드만 탐색하여
HTMLCollection에 담아 반환</li>
  <li>Element.prototype.firstElementChild : 첫 번째 자식 요소 노드 반환</li>
  <li>Element.prototype.lastElementChild : 마지막 자식 요소 노드 반환</li>
</ol>
```

❖ 1-2.DOM

1-2. DOM (노드 탐색)

1-2-1. child node (자식 노드)

```
const $element = document.getElementById('element');  
  
// HTMLCollection 반환, 요소 노드만 포함  
console.log($element.children);  
// 첫 번째 자식 요소 노드 탐색  
console.log($element.firstElementChild);  
// 마지막 자식 요소 노드 탐색  
console.log($element.lastElementChild);
```

❖ 1-2.DOM

1-2. DOM (노드 탐색)

1-2-1. child node (자식 노드)

- Node.prototype.hasChildNodes 메서드를 사용하여 자식 노드 존재 여부를 확인한다.
- 텍스트 노드를 포함하여 자식 노드의 존재를 확인한다.
- 텍스트 노드를 제외한 요소 노드의 존재를 확인하고 싶다면 children.length, childElementCount 등을 사용하여 확인한다.

```
const $empty = document.getElementById('empty');

console.log($node.hasChildNodes()); // true
console.log($element.hasChildNodes()); // true

// 텍스트 노드를 포함하여 확인하므로
// ol 태그 사이를 개행하면 true, 개행하지 않으면 false를 반환한다.
console.log($empty.hasChildNodes());

// 텍스트 노드를 제외한 요소 노드의 존재를 확인하고 싶다면
// children.length, childElementCount 등을 사용하여 확인
console.log(!!$empty.children.length);
console.log(!!$empty.childElementCount);
```

```
<ol id="empty">
</ol>
```


❖ 1-2.DOM

1-2. DOM (노드 탐색)

1-2-2. parent node (부모 노드)

- 부모 노드를 탐색하려면 Node.prototype.parentNode 프로퍼티를 사용한다.
- 텍스트 노드는 DOM 트리의 최종단 노드인 리프 노드(leaf node)이므로 부모가 텍스트 노드인 경우는 없다.

```
<ul id="lists">
  <li class="coffee">커피</li>
  <li class="coke">콜라</li>
  <li class="milk">우유</li>
</ul>
```

```
const $coke = document.querySelector('.coke');
console.log($coke.parentNode);
```

❖ 1-2.DOM

1-2. DOM (노드 탐색)

1-2-3. sibling node (형제 노드)

- **Node.prototype.previousSibling**: 형제 노드 중 자신의 이전 형제 노드(요소 노드, 텍스트 노드)를 탐색하여 반환
- **Node.prototype.nextSibling**: 형제 노드 중 자신의 다음 형제 노드(요소 노드, 텍스트 노드)를 탐색하여 반환

```
<ol id="node">  
  <li>Node.prototype.previousSibling : 형제 노드 중 자신의 이전 형제  
  노드(요소 노드, 텍스트 노드)를 탐색하여 반환</li>  
  <li>Node.prototype.nextSibling : 형제 노드 중 자신의 다음 형제 노드  
(요소 노드, 텍스트 노드)를 탐색하여 반환</li>  
</ol>
```

❖ 1-2.DOM

1-2. DOM (노드 탐색)

1-2-3. sibling node (형제 노드)

```
const $node = document.getElementById('node');

// firstChild 프로퍼티는 요소 노드뿐만 아니라 텍스트 노드 반환 가능
const firstChild = $node.firstChild;
console.log(firstChild);

// firstChild의 다음 형제 노드를 탐색하여 반환
const nextSibling = firstChild.nextSibling;
console.log(nextSibling);

// nextSibling의 이전 형제 노드를 탐색하여 반환
// firstChild -> nextSibling -> previousSibling(firstChild와 동일)
const previousSibling = nextSibling.previousSibling;
console.log(previousSibling);
```

❖ 1-2.DOM

1-2. DOM (노드 탐색)

1-2-3. sibling node (형제 노드)

- **Element.prototype.previousElementSibling** : 형제 요소 노드 중 자신의 이전 형제 요소 노드를 탐색하여 반환
- **Element.prototype.nextElementSibling** : 형제 요소 노드 중 자신의 다음 형제 요소 노드를 탐색하여 반환

```
<ol id="element">
```

```
  <li>Element.prototype.previousElementSibling : 형제 요소 노드 중  
  자신의 이전 형제 요소 노드를 탐색하여 반환</li>
```

```
  <li>Element.prototype.nextElementSibling: 형제 요소 노드 중 자신의  
  다음 형제 요소 노드를 탐색하여 반환</li>
```

```
</ol>
```

❖ 1-2.DOM

1-2. DOM (노드 탐색)

1-2-3. sibling node (형제 노드)

```
const $element = document.getElementById('element');

// firstElementChild 프로퍼티는 요소 노드만 반환
const firstElementChild = $element.firstElementChild;
console.log(firstElementChild);

// firstElementChild의 다음 형제 요소를 탐색하여 반환
const nextElementSibling = firstElementChild.nextElementSibling;
console.log(nextElementSibling);

// nextElementSibling의 이전 형제 요소를 탐색하여 반환
// firstElementChild -> nextElementSibling ->
// previousElementSibling(firstElementChild와 동일)
const previousElementSibling = nextElementSibling.previousElementSibling;
console.log(previousElementSibling);
```