어트리뷰트 노드

어트리뷰트와 DOM 프로퍼티의 대응관계

1. **어트리뷰트 노드**: HTML 어트리뷰트로 지정한 HTML 요소의 초기 상태는 어트리뷰트 노드에서 관리합니다. 어트리뷰트 노드에서 관리하는 어트리뷰트 값은 사용자의 입력에 의해 상태가 변경되어도 변하지 않고 HTML 어트리뷰트로 지정한 HTML 요소의 초기 상태를 그대로 유지합니다. 어트리뷰트 노드가 관리하는 초기 상태 값을 취득하거나 변경하려면 getAttribute.setAttribute 메서드를 사용합니다.
2. **DOM 프로퍼티**: 사용자가 입력한 최신 상태는 HTML 어트리뷰트에 대응하는 요소 노드의 DOM 프로퍼티가 관리합니다. DOM 프러퍼티는 사용자의 입력에 의한 상태 변화에 반응하여 언제나 최신 상태를 유지합니다.

- id 어트리뷰트와 id 프로퍼티는 1:1 대응하며, 동일한 값으로 연동합니다.
- input 요소의 value 어트리뷰트는 value 프로퍼티와 1:1 대응합니다. 하지만 value 어트리뷰트는 초기 상태를 value 프로퍼티는 최신 상태를 갖습니다.
- class 어트리뷰트는 className, classList 프로퍼티와 대응합니다.
- for 어트리뷰트는 htmlFor 프로퍼티와 1:1 대응합니다.
- td 요쇼의 colspan 어트리뷰트는 대응하는 프로퍼티가 존재하지 않습니다.
- textContent 프로퍼티는 대응하는 어트리뷰트가 존재하지 않습니다.
- 어트리뷰트 이름은 대소문자를 구별하지 않지만 대응하는 프로퍼티 키는 카멜 케이스를 따릅니다.

어트리뷰트 노드

- HTML 문서의 구성 요소인 HTML 요소는 여러 개의 어트리뷰트(속성)를 가질 수 있다.
- **Element.prototype.attributes** 프로퍼티로 모든 어트리뷰트 노드를 취득할 수 있다.

```
<input id="user" type="text" value="John">

<script>
// 요소 노드의 attribute 프로퍼티는 요소 노드의 모든 어트리뷰트 노드의 참조가 담긴 NamedNodeMap 객체를 반환한다.
const { attributes } = document.getElementById('user');
```

```
console.log(attributes);
// NamedNodeMap {0: id, 1: type, 2: value, id: id, type: ty
pe, value: value, length: 3}

// 어트리뷰트 값 취득
console.log(attributes.id.value); // user
console.log(attributes.type.value); // text
console.log(attributes.value.value); // John
</script>
```

getAttribute와 setAttribute 메서드를 사용하면 HTML 어트리뷰트 값을 취득하거나 변경할 수 있다.

```
<input id="user" type="text" value="Hello">

<script>
// value 어트리뷰트 값을 취득
const inputValue = $input.getAttribute('value'); // = $inpu
t.value
console.log(inputValue); // Hello

// value 어트리뷰트 값을 변경
$input.setAttribute('value', 'foo'); // = $input.value = 'f
oo'
console.log($input.getAttribute('value')); // foo

//hasAttribute메서드로 어트리뷰트 존재확인하고 removeAttribute로
삭제할 수 있다.
// value 어트리뷰트의 존재 확인
if ($input.hasAttribute('value')) {
  // value 어트리뷰트 삭제
  $input.removeAttribute('value'); //$input.value = "";
}
// value 어트리뷰트가 삭제되었다.
console.log($input.hasAttribute('value')); // false
</script>
```

5초간격 이미지 변환

다운로드 : http://naver.me/xv3qndoj

```
<img src="img/pic1.jpg"  alt="이미지" id="pic">

<style>
#pic {
    width:300px;
    height:auto;
    border-radius:10px;
    box-shadow:1px 1px 2px black;
}
</style>

<script>
const pic = document.querySelector("#pic");//#pic요소선택
let sNum = pic.getAttribute('src').substr(-5,1);
//pic요소의 src속성값의 뒤에서 5번째 1글자를 변수 sNum에 저장
setInterval(gogo,1000);//.gogo함수를 1초에 한번씩 반복 호출
function gogo(){//gogo함수 생성
    if(sNum >= 3) {//sNum변수가 3보다 크거나 같을때 실행
        sNum = 0;//sNum변수의 값을 0로 초기화
    }else{//sNum변수가 3보다 작을때 실행
        sNum++;//sNum변수에 1을 더해 다시 sNum변수에 저장
        pic.setAttribute("src","img/pic"+sNum+".jpg");//pic
요소의 src속성의 이미지경로를 변경하여 이미지를 변환
    }
}
</script>
```

속성 노드의 생성(createAttribute)

createAttribute() 메소드를 사용하여 새로운 속성 노드를 만들 수 있습니다.

만약 같은 이름의 속성 노드가 이미 존재하면, 기존의 속성 노드는 새로운 속성 노드로 대체 됩니다.

이미 존재하는 요소 노드에 속성 노드를 생성하고자 할 때에는 setAttribute() 메소드를 사용할 수 있습니다

```
<h1>속성 노드의 생성</h1>
<p id="text">이 단락에 새로운 속성을 추가할 것입니다.</p>
<button onclick="createNode()">속성 노드 생성!</button>

<script>
function createNode() {
  var text = document.getElementById("text");           //
아이디가 "text"인 요소를 선택함.
  var newAttribute = document.createAttribute("style"); //
새로운 style 속성 노드를 생성함.
  newAttribute.value = "color:red";
  text.setAttributeNode(newAttribute);                  //
해당 요소의 속성 노드로 추가함.
}
</script>
```