스타일 조작

style 프로퍼티

setter와 getter 모두 존재하는 접근자 프로퍼티로서 요소 노드의 인라인 스타일을 취득하거나 추가 또는 변경한다.

```
<div style="color: red">Hello World</div>

<script>
const $div = document.querySelector('div');

// 인라인 스타일 취득
console.log($div.style); // CSSStyleDeclaration { 0: "colo
r", ... }

// 인라인 스타일 변경
$div.style.color = 'blue';

// 인라인 스타일 추가
$div.style.width = '100px';
$div.style.height = '100px';
$div.style.backgroundColor = 'yellow';

//style프로퍼티를 참조하려면 프로퍼티는 카멜 케이스를 따른다.
$div.style.backgroundColor = 'yellow';

//케밥 케이스의 CSS프로퍼티를 그대로 사용하려면 마침표 표기법 대신 대괄
호 표기법을 사용한다.
$div.style['background-color'] = 'yellow';

//CSS 프로퍼티의 값은 반드시 단위를 지정해야 한다.(px, em, % 등)
$div.style.width = '100px';
</script>
```

자바스크립트 콜백

```
<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <title>자바스크립트 콜백</title>
    <style>
        .container{
            display: flex;
        }
        .container > div{
            width: 100px;
            height: 100px;
            margin: 10px;
            background-color: lightcoral;
            border-radius: 50%;
            cursor: pointer;
        }
        .container > div:nth-of-type(2){
            background-color: lightblue;
        }
    </style>
</head>
<body>

<div class="container">
    <div></div>
    <div></div>
</div>

<script>
//onclick, onmouseover, onmouseout 3개의 콜백 함수 형식으로 연결합니다.
document.querySelector('.container > div:nth-of-type(1)').onclick = function(){
        //html요소를 선택하여 자바스크립트 구문을 작성합니다.
    this.style.backgroundColor = 'lightgreen';
}
```

```
document.querySelector('.container > div:nth-of-type(2)').o
nmouseover = function(){
    this.style.borderRadius = 0;
}

document.querySelector('.container > div:nth-of-type(2)').o
nmouseout = function(){
    this.style.borderRadius = '50%';
}
</script>
</body>
</html>
```

window.onload 이벤트

```
<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <title>window.onload 이벤트</title>
    <style>
        img{
            margin: 10px;
            padding: 5px;
            border: 1px solid #ddd;
            cursor: pointer;
            box-shadow: 2px 2px 2px #888;
        }
    </style>
</head>
<body>

    <img src="https://dummyimage.com/150x80/000/ddd" alt
="이미지1">
    <img src="https://dummyimage.com/150x80/666/fff" alt
="이미지2">
```

```
    <script>
        console.log('1번째');
        //onload()는 html 문서 및 링크(경로)로 연결된 CSS, 자바스
크립트, 이미지 등의 파일을 모두 내려받은 후에 호출 됩니다.
        window.onload = function(){
            console.log('2번째');
        }

        console.log('3번째');
    </script>
</body>
</html>
```

onresize 이벤트

```
<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="UTF-8">

    <title>window.onresize 이벤트</title>
    <style>
        img{
            margin: 10px;
            padding: 5px;
            border: 1px solid #ddd;
            cursor: pointer;
            box-shadow: 2px 2px 2px #888;
        }
    </style>
</head>
<body>

    <img src="https://dummyimage.com/200x100/000/fff" alt
="이미지1">

    <script>
```

```
    //onresize를 이용하여 브라우저의 크기가 변했을 때 이벤트를 수행하
도록 구성
        window.onresize = function(){
            console.log('브라우저 크기 - viewport');
            console.log(window.innerWidth +'x'+ window.inne
rHeight);

            console.log('모니터의 크기 - device size');
            console.log(window.screen.width +'x'+ window.sc
reen.height);
        }
    </script>
</body>
</html>
```

클래스 조작

className 프로퍼티는 setter와 getter 모두 존재하는 접근자 프로퍼티로서 HTML요소의 class 어트리뷰트 값을 취득하거나 변경한다.

```
<div class="box red">Hello World</div>
<style>
.box {  width: 100px; height: 100px;  background-color: ant
iquewhite;  }
.red { color: red; }
.blue { color: blue; }
</style>

<script>
const $box = document.querySelector('.box');

// .box 요소의 class 어트리뷰트 값을 취득
console.log($box.className); // 'box red'

// .box 요소의 class 어트리뷰트 값 중에서 'red'만 'blue'로 변경
$box.className = $box.className.replace('red', 'blue');
```

```
// .box 요소의 class 어트리뷰트 정보를 담은 DOMTokenList 객체를 취득
// classList가 반환하는 DOMTokenList 객체는 HTMLCollection과 NodeList와 같이 노드 객체의 상태 변화를 실시간으로 반영하는 살아 있는(live) 객체다.
console.log($box.classList);
// DOMTokenList(2) [length: 2, value: "box blue", 0: "box", 1: "blue"]

// .box 요소의 class 어트리뷰트 값 중에서 'red'만 'blue'로 변경
$box.classList.replace('red', 'blue');

//클래스 추가
$box.classList.add('foo'); // -> class="box red foo"
$box.classList.add('bar', 'baz'); // -> class="box red foo bar baz"

//클래스 제거
$box.classList.remove('foo'); // -> class="box red bar baz"
$box.classList.remove('bar', 'baz'); // -> class="box red"
$box.classList.remove('x'); // -> class="box red"

//인수로 전달한 index에 해당하는 클래스를 반환(0 = 첫번째클래스, 1 = 두번째 클래스)
$box.classList.item(0); // -> "box"
$box.classList.item(1); // -> "red"

//인수로 전달한 문자열과 일치하는 클래스가 포함되어 있으면 true를 없으면 false를 반환
$box.classList.contains('box');  // -> true
$box.classList.contains('blue'); // -> false

//인수로 전달한 문자열과 일치하는 클래스가 존재하면 제거하고, 존재하지 않으면 추가한다.
$box.classList.toggle('foo'); // -> class="box blue foo"
$box.classList.toggle('foo'); // -> class="box blue"
</script>
```

요소에 적용되어 있는 CSS 스타일 참조

- style프로퍼티는 인라인 스타일만 반환한다.
- HTML요소에 적용되어 있는 모든 CSS스타일을 참조해야할 경우 getComputedStyle 메서드를 사용한다.
- window.getComputedStyle(element[, pseudo]) = 첫번째 인수로 전달한 요소노드에 적용되어 있는 평가된 스타일을 반환(최종적으로 적용된 스타일)

```
<div class="box">Box</div>

<style>
body {  color: red;  }
.box {
    width: 100px;  height: 50px;  background-color: cornsil
k;
    border: 1px solid black;
}
.box:before {  content: 'Hello';  }
</style>

<script>
const $box = document.querySelector('.box');

// .box 요소에 적용된 모든 CSS 스타일을 담고 있는 CSSStyleDeclarat
ion 객체를 취득
const computedStyle = window.getComputedStyle($box);
console.log(computedStyle); // CSSStyleDeclaration

// 임베딩 스타일
console.log(computedStyle.width); // 100px
console.log(computedStyle.height); // 50px
console.log(computedStyle.backgroundColor); // rgb(255, 24
8, 220)
console.log(computedStyle.border); // 1px solid rgb(0, 0,
0)
```

```
// 상속 스타일(body -> .box)
console.log(computedStyle.color); // rgb(255, 0, 0)

// 기본 스타일
console.log(computedStyle.display); // block

// 의사 요소 :before의 스타일을 취득한다.
const boxBefore = window.getComputedStyle($box, ':before');
console.log(boxBefore.content); // "Hello"
</script>
```