

# DOM style

DOM(Document Object Model)은 웹 페이지의 구조를 나타내며, 각 요소를 객체로 다룰 수 있는 인터페이스를 제공합니다. 이 중 요소의 외관을 조정하는 데 중요한 역할을 하는 것이 바로 `style` 속성입니다. `style` 속성을 통해 HTML 요소의 스타일을 직접 조작할 수 있습니다.

## style 속성

`style` 속성은 요소의 인라인 스타일을 나타내는 `CSSStyleDeclaration` 객체를 반환합니다. 이 객체를 통해 요소의 다양한 CSS 속성을 읽고 쓸 수 있습니다.

## 기본 사용법

각 HTML 요소는 `HTMLElement` 객체로 표현되며, 이 객체는 `style` 속성을 가지고 있습니다. 이 속성을 사용하여 직접 CSS 속성을 변경할 수 있습니다.

## 예시

```
var myDiv = document.getElementById("myDiv");  
myDiv.style.backgroundColor = "blue";
```

## 자주 사용되는 속성

`style` 속성을 통해 다양한 CSS 속성을 설정할 수 있습니다. 몇 가지 예시는 다음과 같습니다:

- background-color: 요소의 배경색을 설정합니다.
- color: 텍스트 색상을 설정합니다.
- font-size: 텍스트 크기를 설정합니다.
- margin: 요소의 외부 여백을 설정합니다.
- padding: 요소의 내부 여백을 설정합니다.
- border: 요소의 테두리를 설정합니다.

## 주의사항

인라인 스타일로 설정된 CSS 속성은 외부 스타일시트나 내부 스타일시트로 설정된 속성보다 우선순위가 높습니다. 따라서 `style` 속성을 남발하면 유지보수가 어려울 수 있습니다. 필요에 따라 클래스나 외부 스타일시트를 사용하는 것이 좋습니다.

## --- 연습 문제 목록 ---

### 1. 문제 1. 버튼 클릭 시 ` ` 배경색 토글하기

- 요구사항: HTML에 ``과 `

` 요소를 작성합니다. 버튼을 클릭할 때마다 `

`의 배경색을 예를 들어 파란색과 회색 사이에서 번갈아가며 토글하도록 구현하세요.

- 힌트:

- `addEventListener("click", ...)`를 사용하세요.

- `element.style.backgroundColor` 값을 변경합니다.

### 2. 문제 2. 마우스 오버 시 ` `의 폰트 색 변경하기

- 요구사항: HTML에 하나의 `

` 요소를 만들고, 마우스가 요소 위에 올라갈 때 폰트 색상을 빨간색으로, 마우스가 벗어나면 검정색으로 복원하는 기능을 구현하세요.

- 힌트:

- `mouseover`와 `mouseout` 이벤트를 사용하세요.

### 3. 문제 3. 입력 필드 포커스와 블러 시 테두리 변화하기

- 요구사항: `` 요소를 HTML에 만든 후, 이 요소에 포커스가 주어질 경우 테두리 색상을 녹색으로 변경하고, 포커스가 사라지면 원래 색상으로 돌려놓는 기능을 구현하세요.

- 힌트:

- `focus`와 `blur` 이벤트를 활용하세요.

### 4. 문제 4. 버튼으로 텍스트 크기 조절하기

- 요구사항: HTML에 `

` 요소와 두 개의 버튼("확대", "축소")을 만드세요. "확대" 버튼 클릭 시 텍스트 크기가 증가하고, "축소" 버튼 클릭 시 텍스트 크기가 감소하도록 구현하세요.

- 힌트:

- 숫자 값을 변수에 저장하여 매 클릭 시 증가/감소시키고, `style.fontSize`를 변경하세요.

- 초기 폰트 크기를 명시적으로 설정하는 것이 좋습니다.

## 5. 문제 5. 이미지 회전 구현하기

- 요구사항: HTML에 `` 요소와 하나의 버튼을 추가합니다. 버튼을 클릭할 때마다 이미지를 45도씩 회전시키는 기능을 구현하세요.

- 힌트:

- 회전 각도를 저장할 변수를 만들어 매번 45도씩 증가시키세요.

- `style.transform = "rotate(" + angle + "deg);`를 사용합니다.

## 6. 문제 6. 더블 클릭 시 ` ` 배경색 변경하기

- 요구사항: HTML에 `

` 요소를 작성한 후, 해당 요소를 더블 클릭할 때 `

`의 배경색이 노란색으로 변경되도록 구현하세요.

- 힌트:

- `dblclick` 이벤트를 사용하여 스타일을 변경합니다.

## 7. 문제 7. 스크롤 이벤트로 헤더 스타일 바꾸기

- 요구사항: HTML 상단에 고정된 ``를 만들고, 사용자가 페이지를 스크롤할 때 헤더의 배경색이나 투명도가 변화하도록 구현하세요.

- 힌트:

- `window.addEventListener("scroll", ...)`를 활용합니다.

- 예를 들어 스크롤 위치에 따라 배경색의 불투명도를 조절할 수 있습니다.

## 8. 문제 8. 클릭 시 ` ` 투명도 변화 주기

- 요구사항: HTML에 `

` 요소를 작성하고, 해당 요소를 클릭할 때마다 요소의 투명도(opacity)를 예를 들어 1과 0.5 사이에서 토글하도록 구현하세요.

- 힌트:

- 클릭할 때마다 `style.opacity` 값을 조건문으로 변경합니다.

## 9. 문제 9. 우클릭(컨텍스트 메뉴) 시 ` ` 스타일 변경 및 기본 메뉴 방지하기

- 요구사항: HTML에 `

` 요소를 작성한 후, 우클릭(컨텍스트 메뉴) 이벤트가 발생했을 때 기본 메뉴가 나타나지 않도록 하고, 이 이벤트 발생 시 `

`의 글자 색상을 변경하는 기능을 구현하세요.

- 힌트:

- ``contextmenu`` 이벤트를 사용하고, ``event.preventDefault()``로 기본 동작을 막습니다.
- 이후 ``style.color``를 변경하세요.

#### 10. 문제 10. 윈도우 리사이즈에 따른 글자 크기 동적 변경하기

- 요구사항: HTML에 ``<p>`` 요소를 작성하고, 브라우저 창의 크기가 변경될 때 마다 해당 ``<p>``의 글자 크기가 윈도우 너비의 비율에 따라 동적으로 조절되도록 구현하세요.
- 힌트:
  - ``window.addEventListener("resize", ...)``를 사용합니다.
  - 예를 들어, ``window.innerWidth`` 값을 기반으로 폰트 크기를 계산하면 됩니다.

## 2. 문제 정답

2-1

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>문제1 - 버튼 클릭 시 div 배경색 토글</title>
```

```
  <style>
```

```
    #toggleDiv {
```

```
      width: 200px;
```

```
      height: 100px;
```

```
      background-color: blue;
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <button id="toggleBtn">배경색 토글</button>
```

```
  <div id="toggleDiv"></div>
```

```
  <script>
```

```
    const toggleBtn = document.getElementById("toggleBtn");
```

```
    const toggleDiv = document.getElementById("toggleDiv");
```

```
    toggleBtn.addEventListener("click", function () {
```

```
      // 현재 배경색이 blue면 gray로, 그렇지 않으면 blue로 토글
```

```
      if (toggleDiv.style.backgroundColor === "blue") {
```

```
        toggleDiv.style.backgroundColor = "gray";
    } else {
        toggleDiv.style.backgroundColor = "blue";
    }
});
</script>
</body>

</html>
```

2-2

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>문제2 - 마우스 오버 시 p 폰트 색 변경</title>
```

```
</head>
```

```
<body>
```

```
  <p id="text">마우스를 올려보세요.</p>
```

```
  <script>
```

```
    const text = document.getElementById("text");
```

```
    text.addEventListener("mouseover", function () {
```

```
      text.style.color = "red";
```

```
    });
```

```
    text.addEventListener("mouseout", function () {
```

```
      text.style.color = "black";
```

```
    });
```

```
  </script>
```

```
</body>
```

```
</html>
```

2-3

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>문제3 - 입력 필드 포커스와 블러 시 테두리 변화하기</title>
```

```
  <style>
```

```
    #inputField {
```

```
      padding: 10px;
```

```
      border: 2px solid #ccc;
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <input id="inputField" type="text" placeholder="입력하세요.">
```

```
  <script>
```

```
    const inputField = document.getElementById("inputField");
```

```
    inputField.addEventListener("focus", function () {
```

```
      inputField.style.outlineColor = "green";
```

```
    });
```

```
    inputField.addEventListener("blur", function () {
```

```
      inputField.style.outlineColor = "#ccc";
```

```
    });
```



</script>

</body>

</html>

2-4

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>문제4 - 버튼으로 텍스트 크기 조절하기</title>
```

```
  <style>
```

```
    #text {
```

```
      font-size: 16px;
```

```
      /* 초기 폰트 사이즈 */
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <p id="text">이 텍스트의 크기를 조절하세요.</p>
```

```
  <button id="increaseBtn">확대</button>
```

```
  <button id="decreaseBtn">축소</button>
```

```
<script>
```

```
  const text = document.getElementById("text");
```

```
  const increaseBtn = document.getElementById("increaseBtn");
```

```
  const decreaseBtn = document.getElementById("decreaseBtn");
```

```
  let fontSize = 16; // 단위: px
```

```
  increaseBtn.addEventListener("click", function () {
```

```
    fontSize += 2;
```

```
text.style.fontSize = fontSize + "px";  
});
```

```
decreaseBtn.addEventListener("click", function () {  
    fontSize = (fontSize > 2) ? fontSize - 2 : fontSize;  
    text.style.fontSize = fontSize + "px";  
});
```

```
</script>
```

```
</body>
```

```
</html>
```

2-5

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>문제5 - 이미지 회전 구현하기</title>
```

```
  <style>
```

```
    #image {
```

```
      width: 300px;
```

```
      transition: transform 0.3s;
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <!-- 이미지를 위한 경로를 valid한 URL이나 파일 경로로 수정하세요 -->
```

```
  
```

```
  <br>
```

```
  <button id="rotateBtn">회전</button>
```

```
  <script>
```

```
    const image = document.getElementById("image");
```

```
    const rotateBtn = document.getElementById("rotateBtn");
```

```
    let angle = 0;
```

```
    rotateBtn.addEventListener("click", function () {
```

```
      angle += 45;
```

```
image.style.transform = "rotate(" + angle + "deg);
```

```
});
```

```
</script>
```

```
</body>
```

```
</html>
```

2-6

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>문제6 - 더블 클릭 시 div 배경색 변경하기</title>
```

```
  <style>
```

```
    #doubleDiv {
```

```
      width: 200px;
```

```
      height: 150px;
```

```
      background-color: lightblue;
```

```
      line-height: 150px;
```

```
      text-align: center;
```

```
      user-select: none;
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <div id="doubleDiv">더블 클릭하세요</div>
```

```
  <script>
```

```
    const doubleDiv = document.getElementById("doubleDiv");
```

```
    doubleDiv.addEventListener("dblclick", function () {
```

```
      doubleDiv.style.backgroundColor = "yellow";
```

```
    });
```

</script>

</body>

</html>

2-7

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>문제7 - 스크롤 이벤트로 헤더 스타일 바꾸기</title>
```

```
  <style>
```

```
    header {
```

```
      position: fixed;
```

```
      top: 0;
```

```
      left: 0;
```

```
      width: 100%;
```

```
      padding: 20px;
```

```
      background-color: rgba(0, 0, 0, 1);
```

```
      color: white;
```

```
      transition: background-color 0.3s;
```

```
    }
```

```
    .content {
```

```
      margin-top: 80px;
```

```
      height: 2000px;
```

```
      /* 스크롤 생성을 위한 높이 */
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```



```
<header id="pageHeader">헤더 영역</header>
```

```
<div class="content">
```

스크롤하면서 헤더 배경색의 투명도가 변하는 것을 확인해보세요.

```
</div>
```

```
<script>
```

```
const header = document.getElementById("pageHeader");
```

```
window.addEventListener("scroll", function () {
```

```
    let scrollY = window.scrollY;
```

```
    // 스크롤 위치에 따라 투명도를 0.5 ~ 1 사이로 조절
```

```
    let opacity = Math.max(0.5, 1 - scrollY / 400);
```

```
    header.style.backgroundColor = `rgba(0, 0, 0, ${opacity})`;
```

```
});
```

```
</script>
```

```
</body>
```

```
</html>
```

2-8

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>문제8 - 클릭 시 div 투명도 변화 주기</title>
```

```
  <style>
```

```
    #opacityDiv {
```

```
      width: 200px;
```

```
      height: 150px;
```

```
      background-color: coral;
```

```
      opacity: 1;
```

```
      line-height: 150px;
```

```
      text-align: center;
```

```
      user-select: none;
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <div id="opacityDiv">클릭하세요</div>
```

```
  <script>
```

```
    const opacityDiv = document.getElementById("opacityDiv");
```

```
    opacityDiv.addEventListener("click", function () {
```

```
      // 만약 인라인 스타일이 비어있으면 기본값 1로 가정
```

```
let currentOpacity = opacityDiv.style.opacity || 1;
```

```
opacityDiv.style.opacity = (currentOpacity == 1 ? 0.5 : 1);
```

```
});
```

```
</script>
```

```
</body>
```

```
</html>
```

2-9

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>문제9 - 우클릭 시 p 스타일 변경 및 기본 메뉴 방지하기</title>
```

```
</head>
```

```
<body>
```

```
  <p id="contextText">이 문단에서 우클릭 해보세요.</p>
```

```
  <script>
```

```
    const contextText = document.getElementById("contextText");
```

```
    contextText.addEventListener("contextmenu", function (event) {
```

```
      event.preventDefault(); // 기본 우클릭 메뉴 방지
```

```
      contextText.style.color = "purple";
```

```
    });
```

```
  </script>
```

```
</body>
```

```
</html>
```

2-10

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>문제10 - 윈도우 리사이즈에 따른 글자 크기 동적 변경하기</title>
```

```
  <style>
```

```
    #responsiveText {
```

```
      transition: font-size 0.2s;
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <p id="responsiveText">창 크기에 따라 글자 크기가 변합니다.</p>
```

```
  <script>
```

```
    const responsiveText = document.getElementById("responsiveText");
```

```
    function adjustFontSize() {
```

```
      let width = window.innerWidth;
```

```
      // 예시: 창 너비에 따라 폰트 사이즈를 계산 (최소 12px 적용)
```

```
      let size = Math.max(12, width / 50);
```

```
      responsiveText.style.fontSize = size + "px";
```

```
    }
```

```
    window.addEventListener("resize", adjustFontSize);
```

```
// 초기 실행
```

```
adjustFontSize();
```

```
</script>
```

```
</body>
```

```
</html>
```

### 3. 확장 문제

- 문제 1 확장: 일정 시간(예: 3초) 후에 자동으로 원래 색상으로 돌아오는 기능 추가
- 문제 5 확장: 이미지 회전 후, 회전 값을 초기화하는 버튼 구현
- 문제 10 확장: 창 크기에 따라 글자 크기뿐 아니라 전체 레이아웃의 변경을 시도해 보기

#### 4. 확장 문제 정답

4-1

```
<!DOCTYPE html>
```

```
<html lang="ko">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>문제 1 확장 - 3초 후 배경색 복원</title>
```

```
  <style>
```

```
    #toggleDiv {
```

```
      width: 200px;
```

```
      height: 100px;
```

```
      background-color: blue;
```

```
      transition: background-color 0.3s;
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <button id="toggleBtn">배경색 변경</button>
```

```
  <div id="toggleDiv"></div>
```

```
<script>
```

```
  const toggleBtn = document.getElementById("toggleBtn");
```

```
  const toggleDiv = document.getElementById("toggleDiv");
```

```
  const originalColor = "blue";
```

```
  let timer = null;
```



```
toggleBtn.addEventListener("click", function () {
```

```
    // 이전 타이머가 있다면 클리어
```

```
    if (timer) {
```

```
        clearTimeout(timer);
```

```
    }
```

```
    // 배경색 변경 (예: 회색으로 전환)
```

```
    toggleDiv.style.backgroundColor = "gray";
```

```
    // 3초 후 원래 색상으로 복원
```

```
    timer = setTimeout(() => {
```

```
        toggleDiv.style.backgroundColor = originalColor;
```

```
        timer = null;
```

```
    }, 3000);
```

```
});
```

```
</script>
```

```
</body>
```

```
</html>
```

4-2

```
<!DOCTYPE html>

<html lang="ko">

<head>

  <meta charset="UTF-8">

  <title>문제 5 확장 - 이미지 회전 및 리셋</title>

  <style>

    #image {

      width: 300px;

      transition: transform 0.3s;

    }

  </style>

</head>

<body>

  <!-- 이미지를 위한 src 속성을 유효한 이미지 URL로 변경하세요 -->

  

  <br>

  <button id="rotateBtn">회전</button>

  <button id="resetBtn">리셋</button>

  <script>

    const image    = document.getElementById("image");

    const rotateBtn = document.getElementById("rotateBtn");

    const resetBtn  = document.getElementById("resetBtn");

    let angle = 0;

    rotateBtn.addEventListener("click", function() {

      angle += 45;
```

```
    image.style.transform = "rotate(" + angle + "deg)";  
  });
```

```
resetBtn.addEventListener("click", function() {  
    angle = 0;  
    image.style.transform = "rotate(0deg)";  
  });
```

```
</script>
```

```
</body>
```

```
</html>
```

4-3

```
<!DOCTYPE html>
```

```
<html lang="ko">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>문제 10 확장 - 반응형 레이아웃</title>
```

```
<style>
```

```
body {
```

```
margin: 0;
```

```
padding: 0;
```

```
transition: all 0.3s;
```

```
}
```

```
#container {
```

```
margin: 20px;
```

```
transition: all 0.3s;
```

```
}
```

```
header {
```

```
background-color: #333;
```

```
color: white;
```

```
padding: 10px;
```

```
text-align: center;
```

```
transition: all 0.3s;
```

```
}
```

```
#content {
```

```
display: flex;
```

```
flex-direction: column;
```

```
gap: 20px;
```

```
transition: all 0.3s;
```

```
}
```

```
.box {
```

```
  background-color: lightblue;
```

```
  padding: 20px;
```

```
  text-align: center;
```

```
  transition: all 0.3s;
```

```
}
```

```
#responsiveText {
```

```
  transition: font-size 0.3s;
```

```
  margin-top: 20px;
```

```
  text-align: center;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div id="container">
```

```
<header>반응형 레이아웃</header>
```

```
<div id="content">
```

```
<div class="box">Box 1</div>
```

```
<div class="box">Box 2</div>
```

```
<div class="box">Box 3</div>
```

```
</div>
```

```
<p id="responsiveText">창 크기에 따라 글자 크기와 레이아웃이 변합니다.</p>
```

```
</div>
```

```
<script>
```

```
  const responsiveText = document.getElementById("responsiveText");
```

```
  const container      = document.getElementById("container");
```

```
const content = document.getElementById("content");
```

```
function adjustLayout() {
```

```
    const width = window.innerWidth;
```

```
    // 창 너비에 따른 폰트 크기 조정 (최소 12px)
```

```
    const fontSize = Math.max(12, width / 50);
```

```
    responsiveText.style.fontSize = fontSize + "px";
```

```
    // 레이아웃 조정: 창 너비가 600px 이상이면 가로 레이아웃, 작으면 세로 레이아웃
```

```
    if (width > 600) {
```

```
        content.style.flexDirection = "row";
```

```
        container.style.margin = "40px";
```

```
    } else {
```

```
        content.style.flexDirection = "column";
```

```
        container.style.margin = "20px";
```

```
    }
```

```
}
```

```
window.addEventListener("resize", adjustLayout);
```

```
adjustLayout(); // 초기 호출
```

```
</script>
```

```
</body>
```

```
</html>
```