

자바스크립트 과제

과제 : 내용물이 작성되어있는 새로운 팝업창을 출력

- 조건

- 버튼 이름은 팝업창 열기로 지정
- 버튼을 클릭하면 팝업창 오픈되도록(popup.html)
- 가로450px, 세로 300px 크기로 팝업이 열리도록
- 팝업이 열리는 위치는 화면의 정 가운데
- 열린 팝업창의 콘솔창에서는 0000년 00월 00일 시:분:초가 1초 단위로 나타나도록
- 열린 팝업창의 내용은 태그를 이용하여 이미지가 출력되도록(popup.html에 구현)

자바스크립트 과제

- 힌트

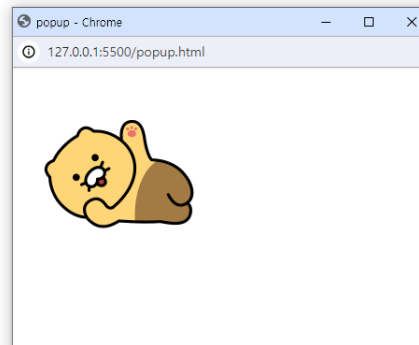
- 팝업이 열리는 위치 구하는 방법 :

```
var _left = (window.screen.width - 지정된 width) /2;
```

```
var _top = (window.screen.height - 지정된 height) /2;
```

```
window.open('ooo.html', '팝업창 열기', '여기에 width,  
height 입력 후 left, top도 추가')
```

자바스크립트 과제 완성화면



자바스크립트 과제 완성소스

open.html

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>240516과제</title>
</head>
<body>
  <button onclick="popup()">팝업창 열기</button>
  <script>
    function popup(){
      var _width = 450;
      var _height = 300;
      var _left = (window.screen.width - 450) / 2;
      var _top = (window.screen.height - 300) / 2;
      window.open('popup.html', '팝업창 열기', 'width='+_width+', height='+_height+', left ='+'_left+',top='+_top);
    }
  </script>
</body>
</html>
```

자바스크립트 과제 완성소스

popup.html

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>popup</title>
</head>
<body>
  <script src="240516_2.js"></script>
  
</body>
</html>
```

12장

문서 객체 모델과 이벤트 다루기

12.1 문서 객체 모델 이해하기

12.2 노드 선택하기

12.3 노드 조작하기

12.4 노드 추가/삭제하기

12.5 폼 조작하기

12.6 이벤트 다루기

12.7 이벤트 객체와 this

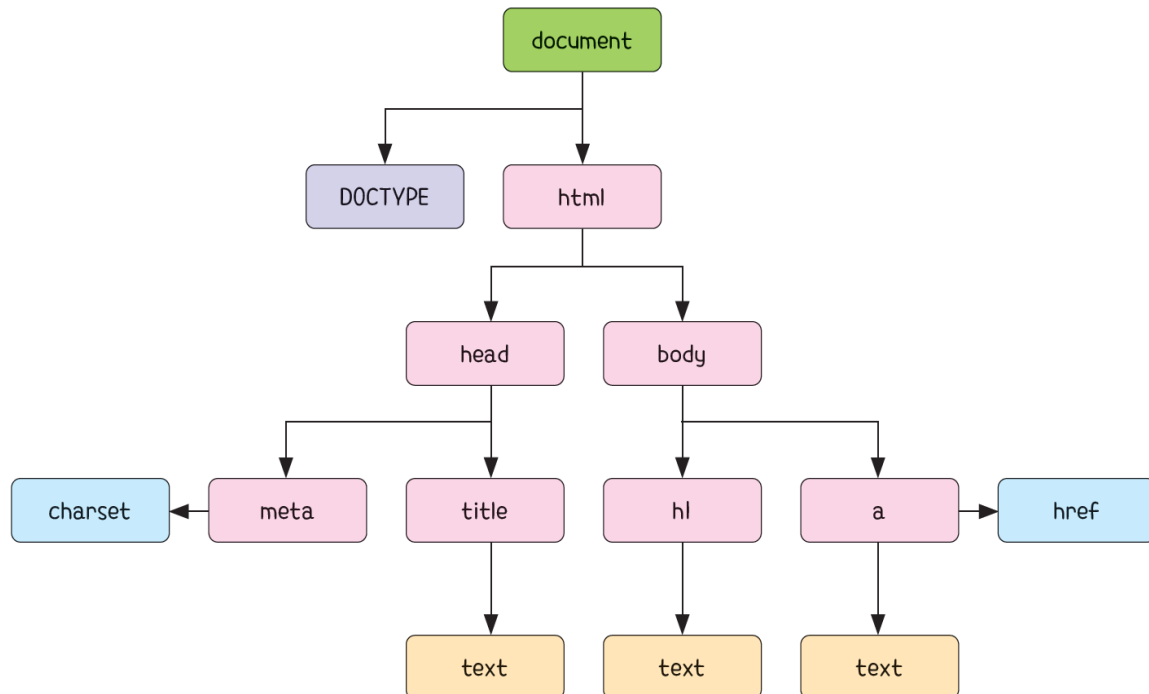
12.1 문서 객체 모델 이해하기

1. 문서 객체 모델이 생성되는 방식

- 문서 객체 모델(DOM, Document Object Model) : 웹 브라우저에 표시되는 문서를 자바스크립트가 이해할 수 있도록 객체화한 모델 구조

그림 12-2 DOM 트리

- DOM 트리 :
문서 객체 모델이
가지는 트리(tree) 구조



12.1 문서 객체 모델 이해하기

2. 노드 타입 살펴보기

(총 9가지가 있지만 HTML5 기반에서 사용되는 5가지 노드만 숙지)

- 문서 노드(Node.DOCUMENT_NODE) : 최상위 document 객체의 노드 타입
- 요소 노드(Node.ELEMENT_NODE) : 요소의 노드 타입
예 h1, p 태그
- 속성 노드(Node.ATTRIBUTE_NODE) : 속성의 노드 타입
예 href, src 속성
- 텍스트 노드(Node.TEXT_NODE) : 텍스트에 해당하는 노드 타입
- 주석 노드(Node.COMMENT_NODE) : 주석에 해당하는 노드 타입

12.2 노드 선택하기

1. 속성으로 노드 선택하기

- 모든 요소 탐색 속성
 - parentNode : 부모 노드를 반환
 - childNodes : 모든 자식 노드를 반환
 - firstChild : 첫 번째 자식 노드를 반환
 - lastChild : 마지막 자식 노드를 반환
 - previousSibling : 이전 형제 노드를 반환
 - nextSibling : 다음 형제 노드를 반환

12.2 노드 선택하기

- 요소 노드 탐색 속성
 - `parentElement` : 부모 요소 노드를 반환
 - `children` : 자식 요소 노드를 반환
 - `firstElementChild` : 첫 번째 자식 요소 노드를 반환
 - `lastElementChild` : 마지막 자식 요소 노드를 반환
 - `previousElementSibling` : 이전 요소 노드를 반환
 - `nextElementSibling` : 다음 요소 노드를 반환

12.2 노드 선택하기

1. 속성으로 노드 선택하기 예제

- firstChild 속성을 사용해 첫 번째 자식 요소 노드로 이동

```
document.firstChild; // <DOCTYPE html>  
document.firstChild; // html
```

- 노드 탐색속성 연속 사용

```
document.childNodes[1].firstElementChild.firstElementChild.nextElementSibling;
```

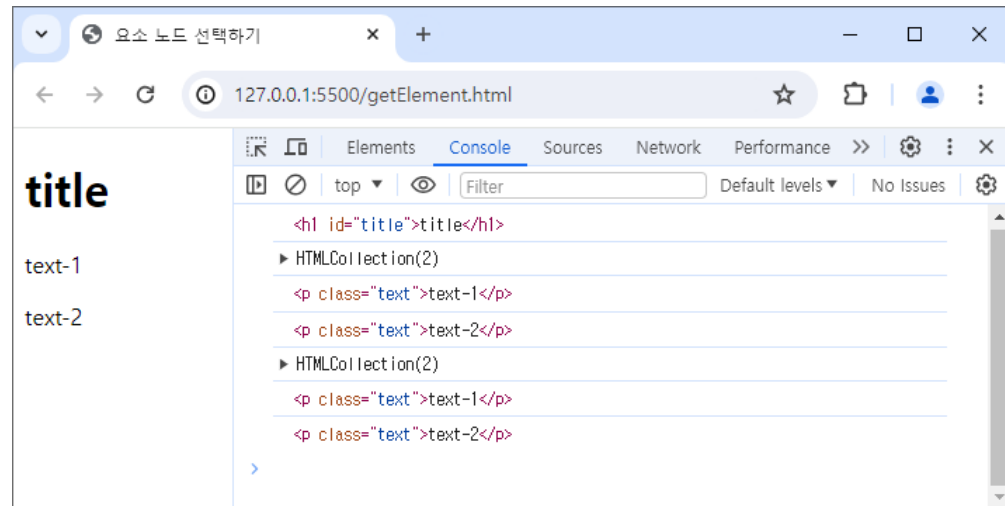
12.2 노드 선택하기

2. 메서드로 노드 선택하기

- 노드의 타입을 가리지 않고 모든 노드를 이동하여 선택가능 장점
- 속성값과 태그명 사용하기 - get 메서드
 - getElementById(<id 속성값>) : id 속성값과 일치하는 요소 노드를 1개만 선택
 - getElementsByClassName(<class 속성값>) : class 속성값과 일치하는 요소 노드를 모두 선택
 - getElementsByTagName(<태그명>) : 태그명과 일치하는 요소 노드를 모두 선택

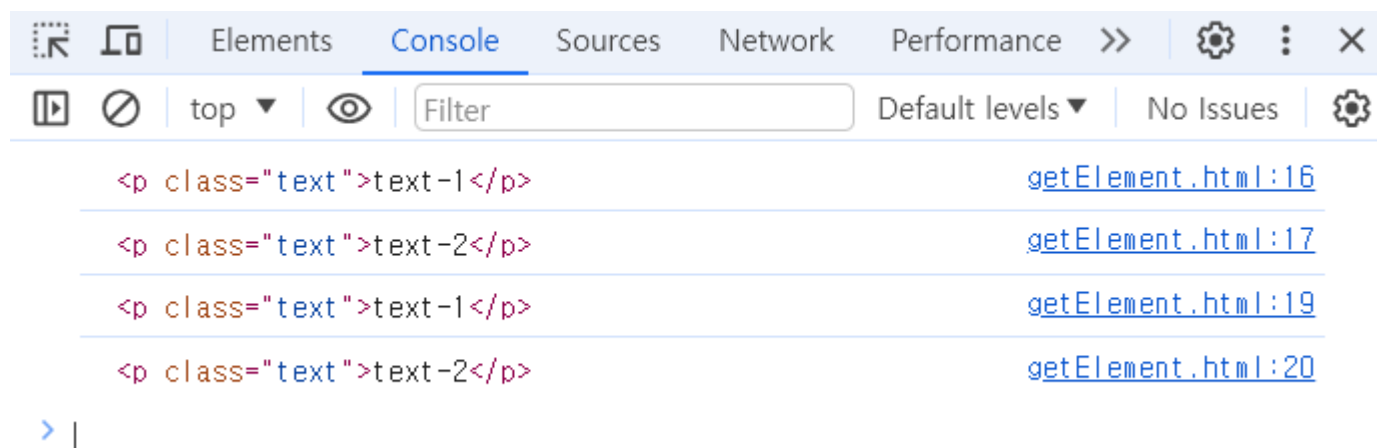
12.2 노드 선택하기

```
<body>
  <h1 id="title">title</h1>
  <p class="text">text-1</p>
  <p class="text">text-2</p>
  <script>
    // id 속성값이 title인 요소 노드 선택하기
    const el = document.getElementById("title");
    console.log(el);
    // class 속성값이 text인 요소 노드 모두 선택하기
    const classEl = document.getElementsByClassName("text");
    console.log(classEl);
    for(let el of classEl){
      console.log(el);
    }
    // p 태그에 해당하는 요소 노드 모두 선택하기
    const tagEls = document.getElementsByTagName("p");
    console.log(tagEls);
    for(let el of tagEls){
      console.log(el);
    }
  </script>
</body>
```



12.2 노드 선택하기

```
<script>
  const classEl = document.getElementsByClassName("text");
  console.log(classEl[0]);
  console.log(classEl[1]);
  const tagEls = document.getElementsByTagName("p");
  console.log(tagEls[0]);
  console.log(tagEls[1]);
</script>
```



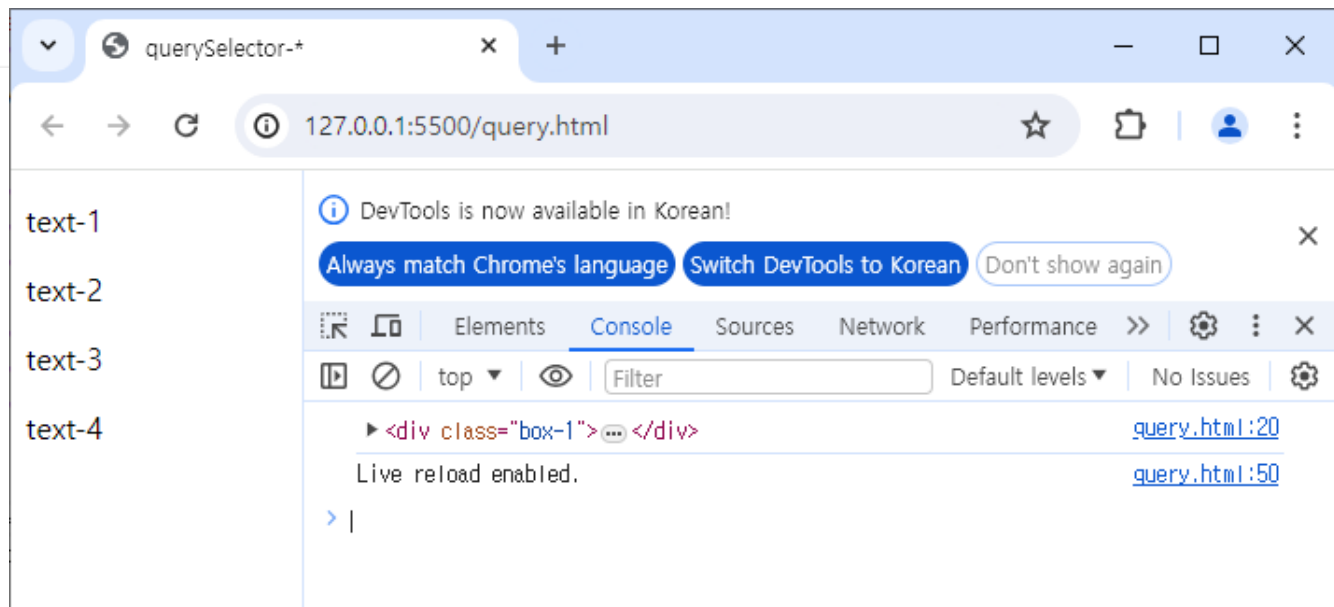
12.2 노드 선택하기

2. 메서드로 노드 선택하기

- CSS 선택자 사용하기 - query 메서드
 - `querySelector(<CSS 선택자>)` : 매개변수로 넘어오는 CSS 선택자에 해당하는 노드를 1개만 선택
 - `querySelectorAll(<CSS 선택자>)` : 매개변수로 넘어오는 CSS 선택자에 해당하는 노드를 모두 선택(노드 2개이상)

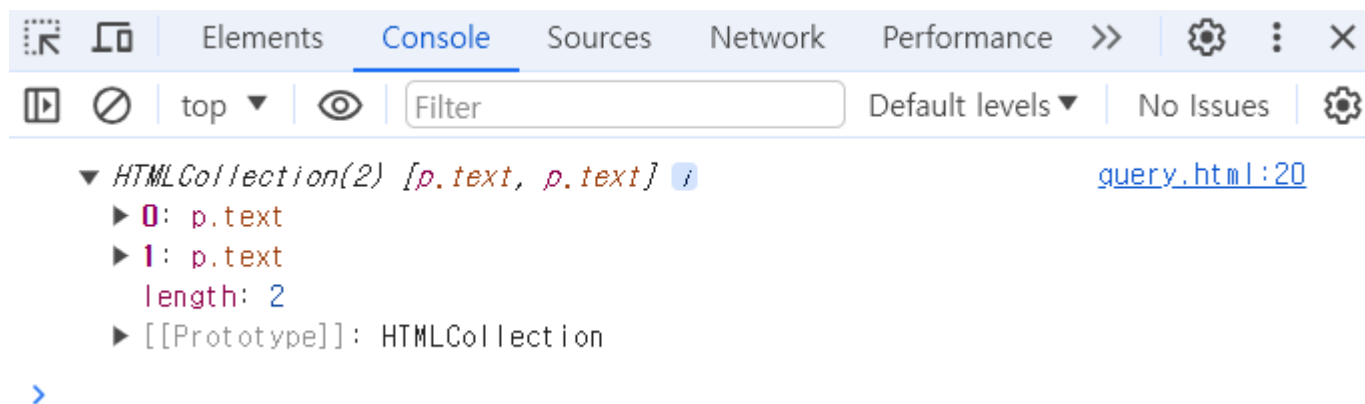
12.2 노드 선택하기

```
<body>
  <div class="box-1">
    <p class="text">text-1</p>
    <p class="text">text-2</p>
  </div>
  <div class="box-2">
    <p class="text">text-3</p>
    <p class="text">text-4</p>
  </div>
  <script>
    const el = document.querySelector(".box-1");
    console.log(el);
  </script>
</body>
```



12.2 노드 선택하기

```
<script>  
  const el = document.getElementsByClassName("box-1")[0].children;  
  console.log(el);  
</script>
```



12.2 노드 선택하기

12.3 노드 조작하기

1. 콘텐츠 조작하기

- `textContent` : 노드 요소의 모든 텍스트에 접근
- `innerText` : 노드 요소의 텍스트 중 웹 브라우저에 표시되는 텍스트에만 접근
- `innerHTML` : 노드 요소의 텍스트 중 HTML 태그를 포함한 텍스트에만 접근

12.3 노드 조작하기

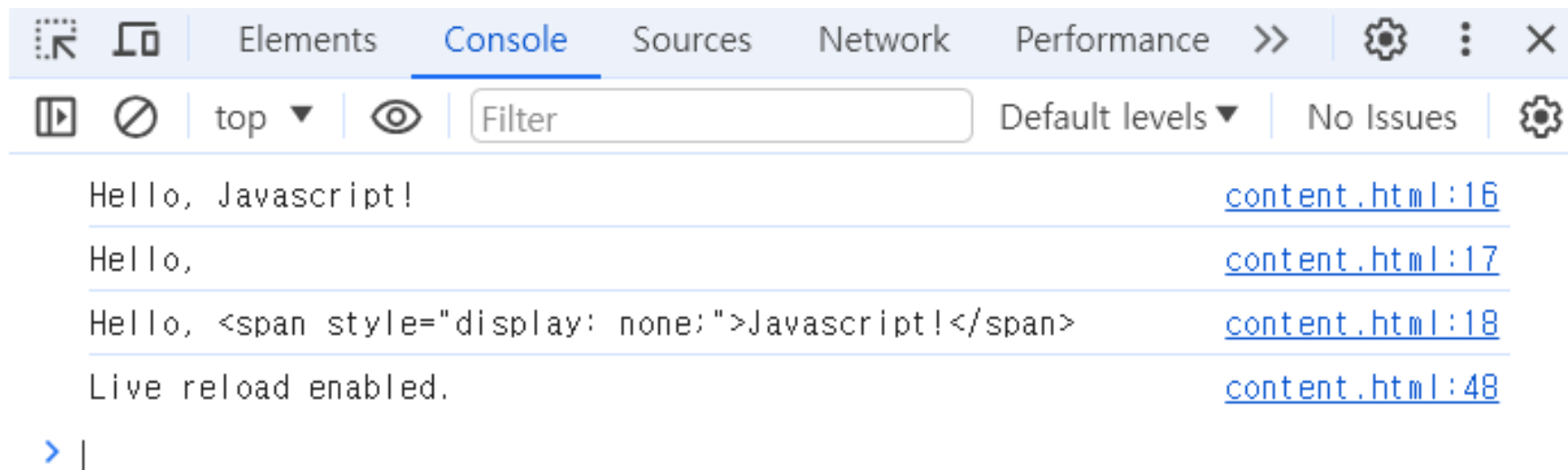
1. 콘텐츠 조작하기 예제

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>콘텐츠 조작</title>
</head>
<body>
  <p id="title">Hello, <span style="display: none;">Javascript!</span></p>
  <script>
    document.getElementById("title").textContent; // Hello, Javascript!
    document.getElementById("title").innerText; // Hello,
    document.getElementById("title").innerHTML; // Hello, <span style="display: none;">Javascript!</span>

    console.log(document.getElementById("title").textContent);
    console.log(document.getElementById("title").innerText);
    console.log(document.getElementById("title").innerHTML);
  </script>
</body>
</html>
```

12.3 노드 조작하기

1. 콘텐츠 조작하기 예제



12.3 노드 조작하기

1. 콘텐츠 조작하기(변경) 예제

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>DOM Modify Set</title>
</head>
<body>
  <p id="textContent"></p>
  <p id="innerText"></p>
  <p id="innerHTML"></p>
  <script>
    document.querySelector("#textContent").textContent = `<strong>textContent</strong> 속성`;
    document.querySelector("#innerText").innerText = `<strong>innerText</strong> 속성`;
    document.querySelector("#innerHTML").innerHTML = `<strong>innerHTML</strong> 속성`;
  </script>
</body>
</html>
```

12.3 노드 조작하기

1. 콘텐츠 조작하기(변경) 예제

DOM Modify Set

127.0.0.1:5500/content_set.html

textContent 속성

innerText 속성

innerHTML 속성

DevTools is now available in Korean!

Always match Chrome's language Switch DevTools to Korean Don't show again

Elements Console Sources Network Performance

```
<!DOCTYPE html>
<html lang="ko">
  <head>
  </head>
  <body>
    <p id="textContent"><strong>textContent</strong> 속성</p>
    <p id="innerText"><strong>innerText</strong> 속성</p>
    <p id="innerHTML" == $0
      <strong>innerHTML</strong>
      " 속성 "
    </p>
    <script>
    </script>
    <!-- Code injected by live-server -->
    <script>
    </script>
  </body>
</html>
```

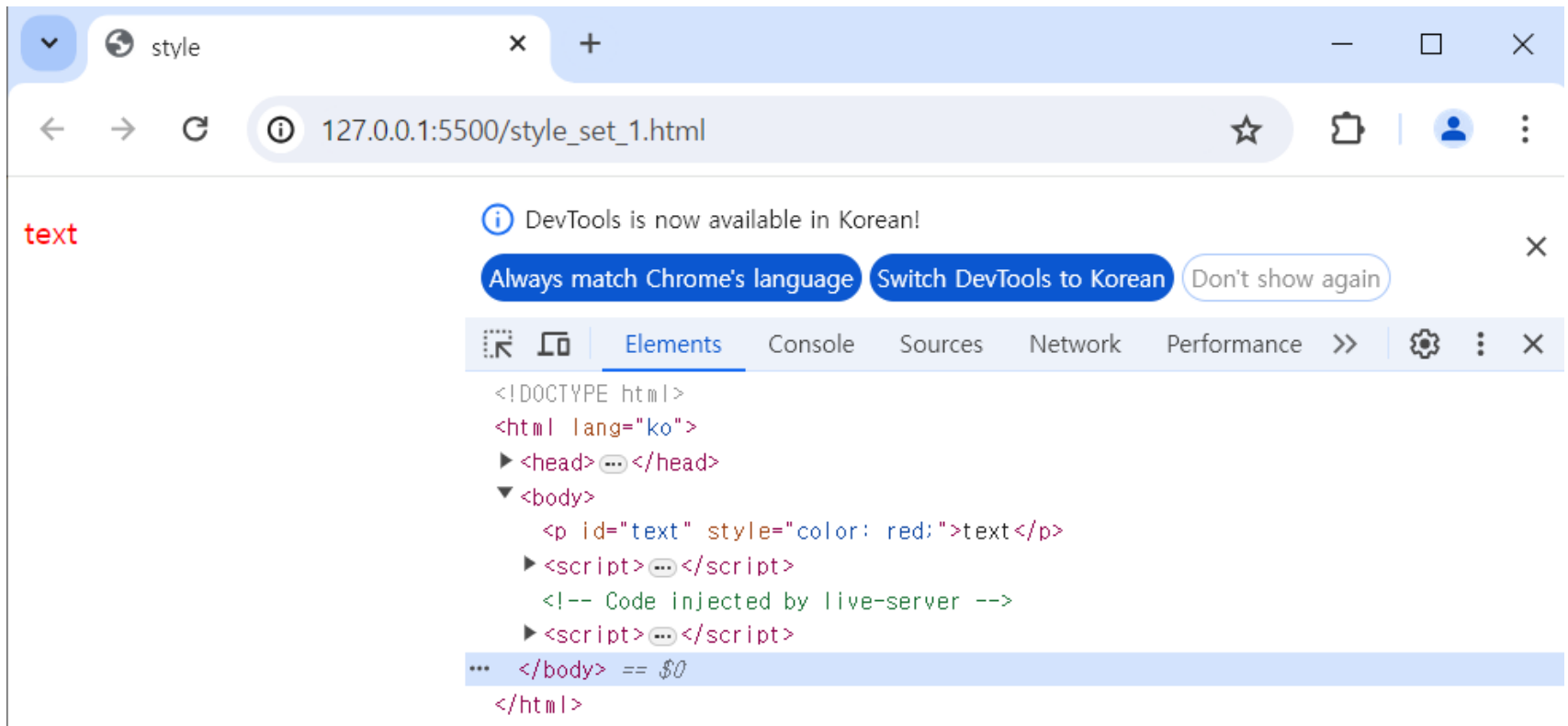
12.3 노드 조작하기

2. 스타일 조작하기

형식 <노드>.style.<css 속성명> = <속성값>;

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>style</title>
</head>
<body>
  <p id="text">text</p>
  <script>
    const pEl = document.querySelector("p"); // 노드 선택하기
    pEl.style.color = "red";
  </script>
</body>
</html>
```


12.3 노드 조작하기



12.3 노드 조작하기

2. 스타일 조작하기 주의점

- CSS속성 중 **background-color**속성과 같이 속성명에 대시(-)가 있는 속성은 자바스크립트에서는 뉘셈 연산자로 인식하여 카멜 표기법으로 변경해서 작성

```
<script>  
  const pEl = document.querySelector("p"); // 노드 선택하기  
  pEl.style.backgroundColor = "#ff0000";  
  pEl.style.fontSize = "20px";  
  pEl.style.color = "#ffffff";  
</script>
```

12.3 노드 조작하기

3. 클래스 속성 조작하기

```
<script>
  const pEl = document.querySelector("p"); // 노드 선택하기
  pEl.style.backgroundColor = "#ff0000";
  pEl.style.fontSize = "20px";
  pEl.style.color = "#ffffff";
</script>
```



```
<style>
  .active{
    color: ■ red;
    font-size: 20px;
    font-weight: bold;
    line-height: 1;
  }
</style>
</head>
<body>
<p class="active">text</p>
</body>
```

12.3 노드 조작하기

3. 클래스 속성 조작하기

형식 <노드>.classList.add("class 속성값"); // 추가

<노드>.classList.remove("class 속성값"); // 삭제

<노드>.classList.toggle("class 속성값"); // 추가와 삭제 반복

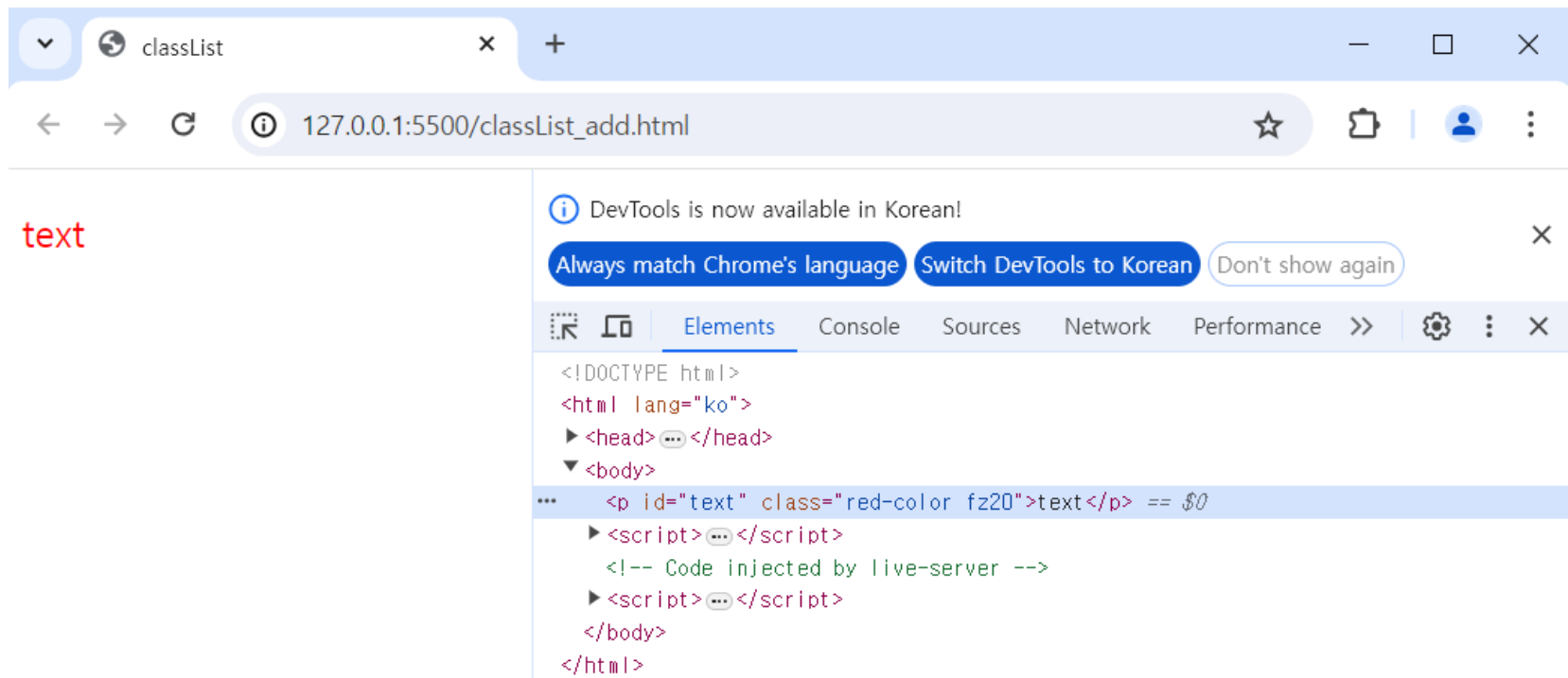
12.3 노드 조작하기

3. 클래스 속성 조작하기 예제

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>classList</title>
  <style>
    .red-color{
      color: ■red;
    }
    .fz20{
      font-size:20px;
    }
  </style>
</head>
<body>
  <p id="text">text</p>
  <script>
    const pEl = document.querySelector("#text"); // 노드 선택하기
    pEl.classList.add("red-color");
    pEl.classList.add("fz20");
  </script>
</body>
</html>
```

12.3 노드 조작하기

3. 클래스 속성 조작하기 예제



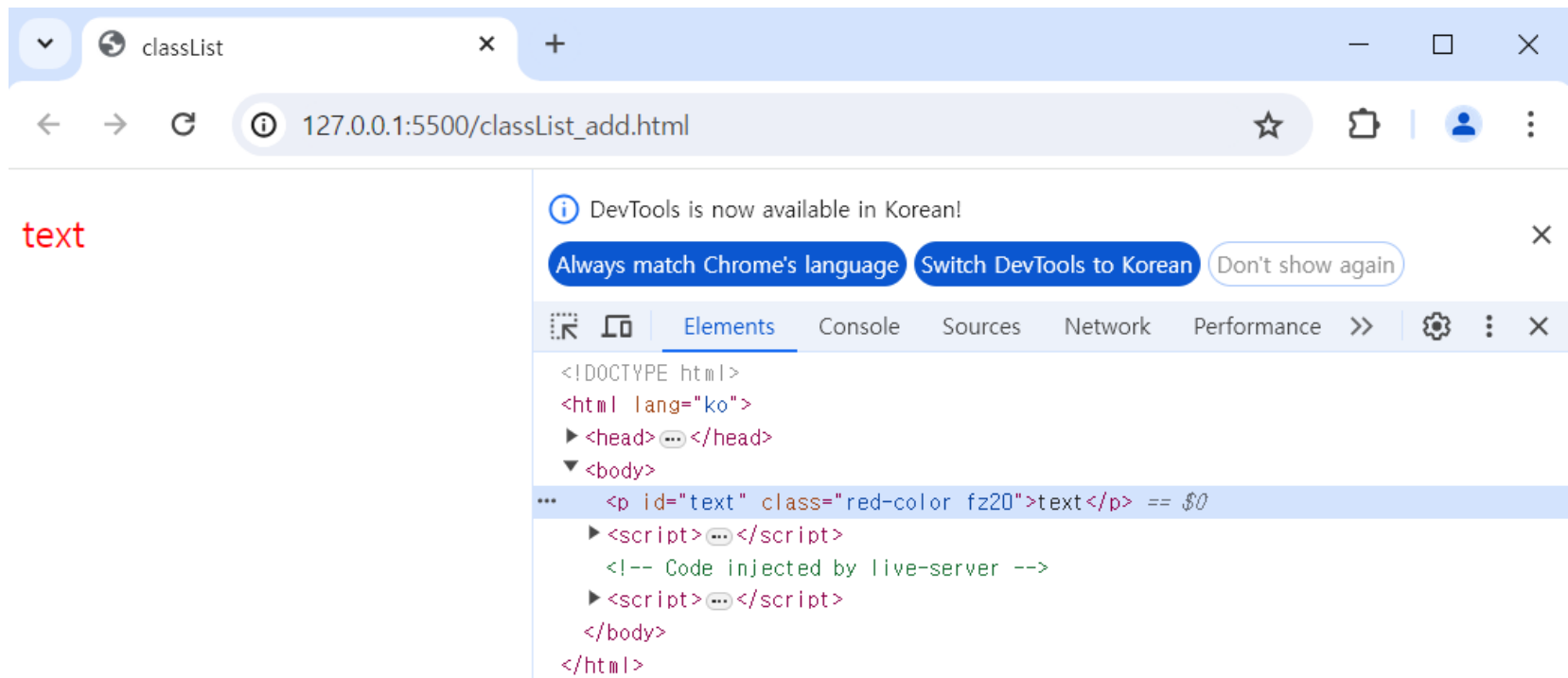
12.3 노드 조작하기

3. 클래스 속성 조작하기 예제

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>classList</title>
  <style>
    .red-color{
      color: red;
    }
    .fz20{
      font-size: 20px;
    }
  </style>
</head>
<body>
  <p id="text">text</p>
  <script>
    const pEl = document.querySelector("#text"); // 노드 선택하기
    pEl.classList.add("red-color", "fz20");
  </script>
</body>
</html>
```

12.3 노드 조작하기

3. 클래스 속성 조작하기 예제



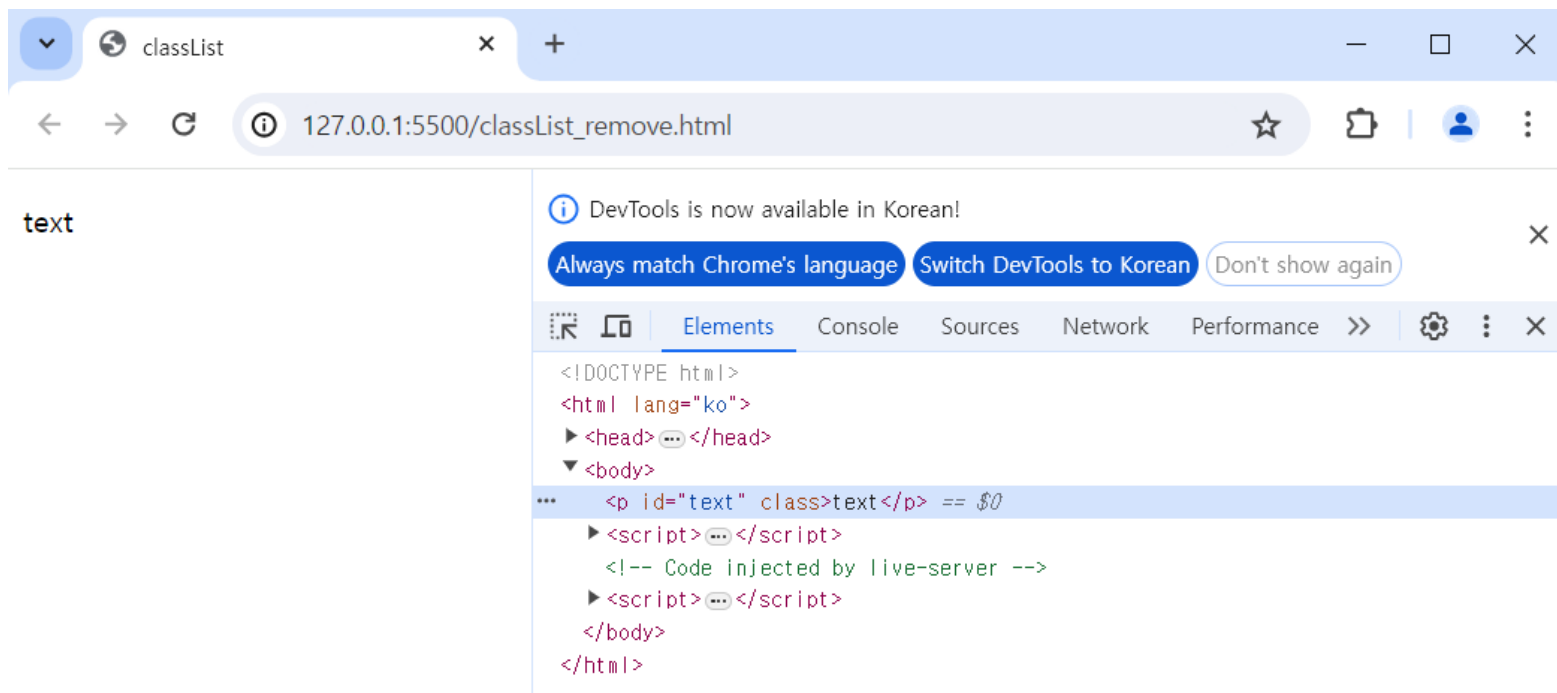
12.3 노드 조작하기

3. 클래스 속성 조작하기 예제

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>classList</title>
  <style>
    .red-color{
      color: ■red;
    }
    .fz20{
      font-size:20px;
    }
  </style>
</head>
<body>
  <p id="text" class="red-color fz20">text</p>
  <script>
    const pEl = document.querySelector("#text");
    pEl.classList.remove("red-color", "fz20"); // 삭제
  </script>
</body>
</html>
```

12.3 노드 조작하기

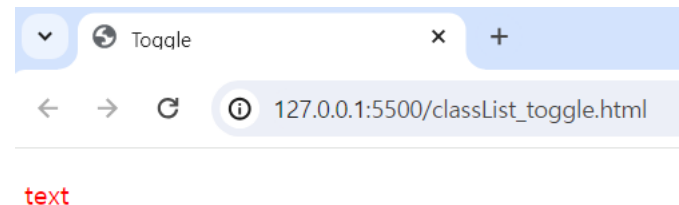
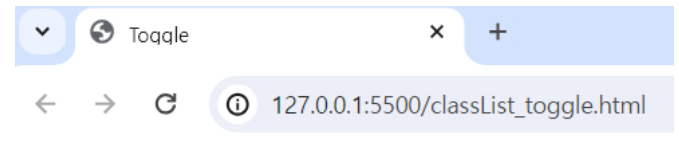
3. 클래스 속성 조작하기 예제



12.3 노드 조작하기

3. 클래스 속성 조작하기 예제

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Toggle</title>
  <style>
    .red-color{
      color: red;
    }
  </style>
</head>
<body>
  <p id="text">text</p>
  <script>
    const pEl = document.querySelector("#text"); // 노드 선택
    // 1초마다 toggle 메서드 반복 실행
    setInterval(function(){
      pEl.classList.toggle("red-color");
    }, 1000);
  </script>
</body>
</html>
```



12.3 노드 조작하기

4. 데이터 속성 조작하기

- data-* 속성
 - 사용자 정의(custom) 속성
 - dataset 속성을 사용해 조작(dataset 속성은 HTML 문서에서 data-* 속성을 가져오거나 지정)

12.3 노드 조작하기

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>dataset 속성</title>
</head>
<body>
  <button data-cnt="10">가방 구매</button>
  <button data-cnt="0">신발 구매</button>
  <script>
    const buttonEls = document.querySelectorAll("button");
    buttonEls.forEach((el) => {
      console.log(el.dataset);
    })
  </script>
</body>
</html>
```

dataset 속성

127.0.0.1:5500/dataset.html

가방 구매

신발 구매

DevTools is now available in Korean!

Always match Chrome's language Switch DevTools to Korean Don't show again

Elements Console Sources Network >>

top Filter Default levels No Issues

DOMStringMap dataset.html:15
cnt: "10"
[[Prototype]]: DOMStringMap

DOMStringMap dataset.html:15
cnt: "0"
[[Prototype]]: DOMStringMap

Live reload enabled. dataset.html:46

12.3 노드 조작하기

5. 메서드로 속성 조작하기

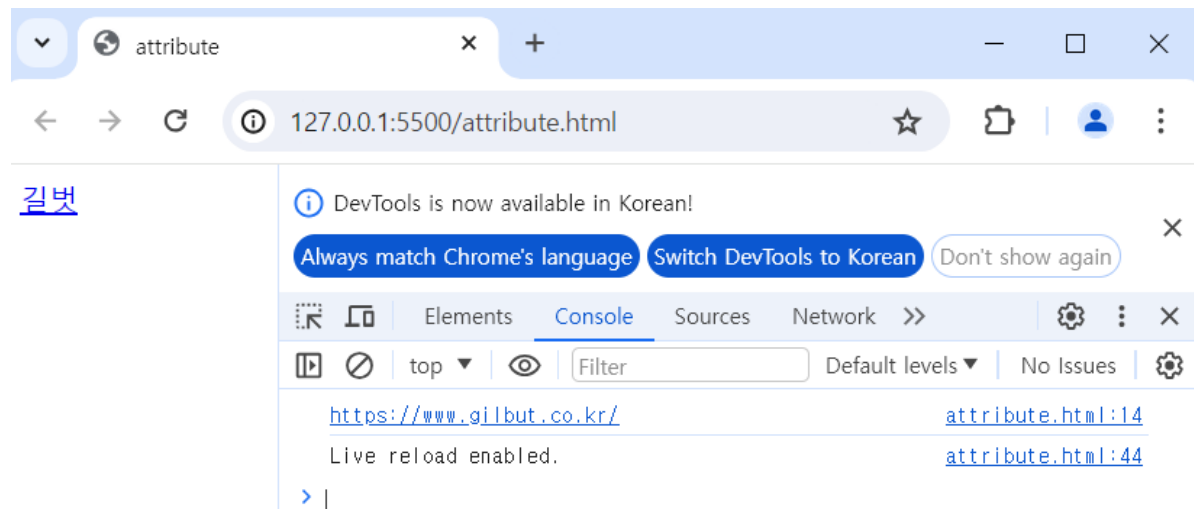
형식 <노드>.getAttribute("속성명"); // 속성값 가져오기

<노드>.setAttribute("속성명", "속성값"); // 속성값 설정

<노드>.removeAttribute("속성명"); // 속성 삭제

12.3 노드 조작하기

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>attribute</title>
</head>
<body>
  <a href="https://www.gilbut.co.kr/">길벗</a>
  <script>
    const aEl = document.querySelector("a");
    const href = aEl.getAttribute("href");
    console.log(href);
  </script>
</body>
</html>
```



12.3 노드 조작하기

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>attribute</title>
</head>
<body>
  <a href="https://www.gilbut.co.kr/">길벗</a>
  <script>
    const aEl = document.querySelector("a");
    const href = aEl.getAttribute("href");
    aEl.setAttribute("href", "https://www.sucoding.kr");
    aEl.innerText = "수코딩";
    console.log(href);
  </script>
</body>
</html>
```

수코딩

attribute

127.0.0.1:5500/attribute.html

수코딩

DevTools is now available in Korean!

Always match Chrome's language Switch DevTools to Korean Don't show again

Elements Console Sources Network

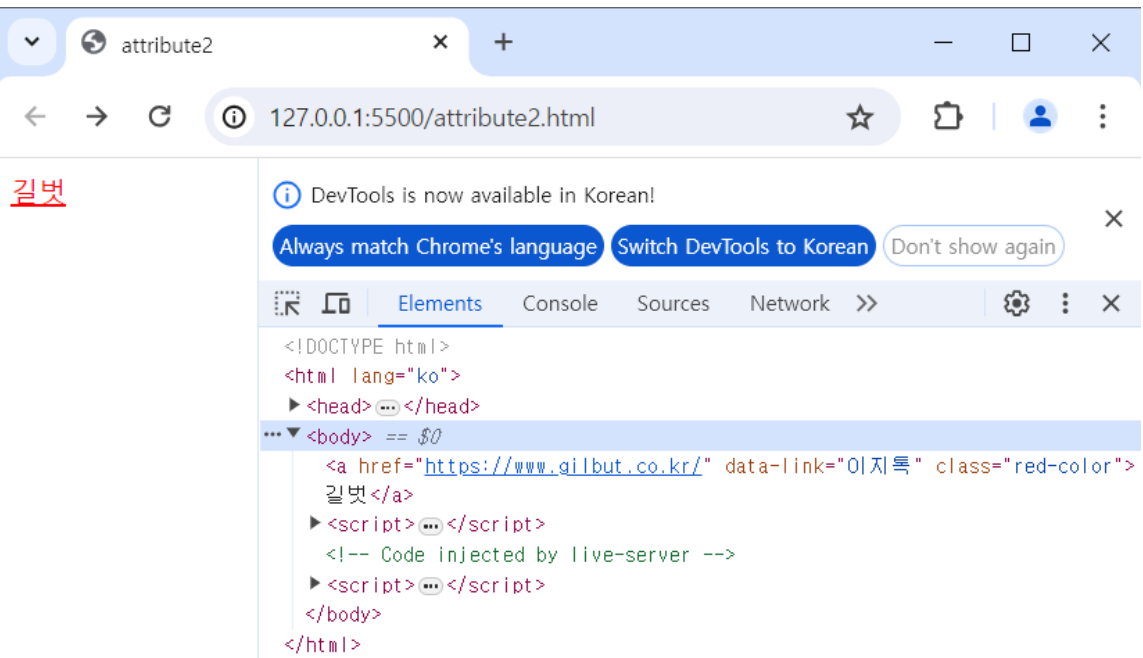
```
<!DOCTYPE html>
<html lang="ko">
  <head>
  </head>
  <body>
    <a href="https://www.sucoding.kr">수코딩</a>
    <script>
    </script>
    <!-- Code injected by live-server -->
    <script>
    </script>
  </body>
</html>
```


12.3 노드 조작하기

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>attribute</title>
</head>
<body>
  <a href="https://www.gilbut.co.kr/">길벗</a>
  <script>
    const aEl = document.querySelector("a");
    const href = aEl.getAttribute("href");
    aEl.setAttribute("href", "https://www.sucoding.kr");
    aEl.innerText = "수코딩";
    aEl.setAttribute("target", "_blank");
  </script>
</body>
</html>
```

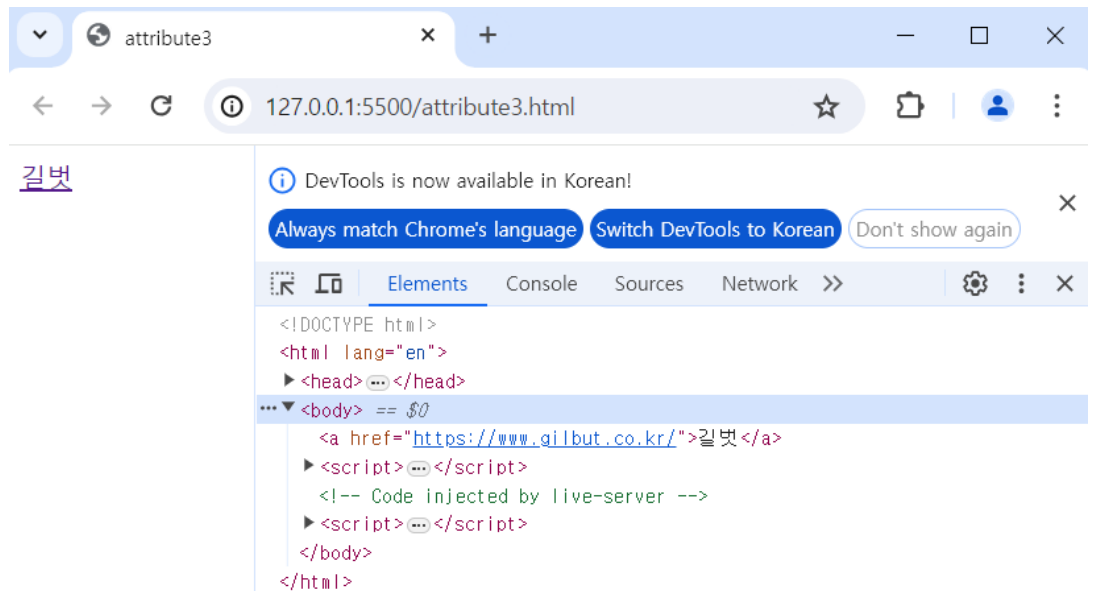
12.3 노드 조작하기

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>attribute2</title>
  <style>
    .red-color{
      color: ■red;
    }
  </style>
</head>
<body>
  <a href="https://www.gilbut.co.kr/" data-link="길벗">길벗</a>
  <script>
    const aEl = document.querySelector("a");
    aEl.setAttribute("data-link", "이지톡");
    aEl.setAttribute("class", "red-color");
  </script>
</body>
</html>
```



12.3 노드 조작하기

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>attribute3</title>
  <style>
    .red-color{
      color: ■red;
    }
  </style>
</head>
<body>
  <a href="https://www.gilbut.co.kr/" class="red-color">길벗</a>
  <script>
    const aEl = document.querySelector("a");
    aEl.removeAttribute("class"); // class 속성 삭제
  </script>
</body>
</html>
```



과제

과제 : 저번주 목요일 과제의 popup.html 이용하여 과제 수행

```
<body>
  <script src="clork.js"></script>
  <div id="clork"></div>
  <script>

  </script>
  
</body>
```

- 조건

- 콘솔창에서 0000년 00월 00일 시:분:초가 1초단위로 나타나던 시계를 div를 id값으로 불러와서 div에 표시
- Toggle()메서드를 이용하여 1초마다 toggle()메서드 반복 실행(toggle예제 활용)

과제 실행 화면

2024년 5월 21일 0:22:32



2024년 5월 21일 0:22:43

