01

DOM 모델의 개념 및 활용

학습내용

01 DOM (Document Object Model)의 개념 이해

02 DOM 트리 구조와 노드의 종류

03 DOM 요소 선택하기

04 DOM 요소 조작하기

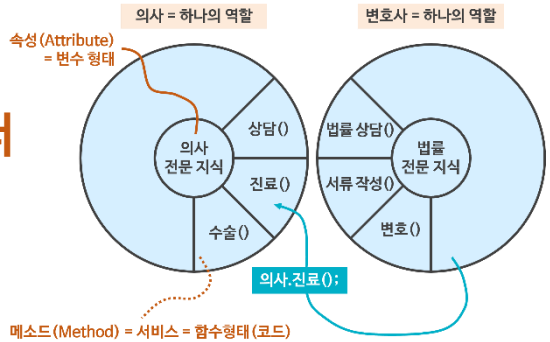
학습목표

- DOM의 개념과 역할을 이해하고 설명할 수 있다.
- DOM 트리 구조와 노드의 종류를 구분할 수 있다.
- 다양한 DOM 요소 선택 방법을 적용할 수 있다.
- JavaScript로 DOM 요소를 조작하여 동적 웹 콘텐츠를 구현할 수 있다.

지난 주차 복습

객체지향 프로그래밍의 핵심 개념

- Object
  = 단일 책임 = **단일 역할**
  = **관련된 다수의 기능 + 관련된 다수의 데이터**
  = 독립적 실행 단위 = Code + Data



JavaScript의 특징

- JavaScript 언어는 멀티 패러다임을 지원

지난 주차 복습

절차지향 vs 객체지향 vs 함수형 프로그래밍의 비교

항목	절차지향 프로그래밍 (Procedural)	객체지향 프로그래밍 (OOP)	함수형 프로그래밍 (Functional)
핵심 단위	함수(절차, 알고리즘)	객체(데이터+행동)	순수 함수, 고차 함수
중심 개념	순차적 처리 흐름	역할, 캡슐화, 상속	불변성, 상태 없음
데이터와 로직	분리	통합	완전 분리
재사용성	낮음(복사)	높음(상속, 캡슐화)	매우 높음(함수 조합)
변화 대응력	약함	보통	높음(데이터 독립)
대표 언어	C, Pascal	Java, Python, C++	Haskell, JS(ES6+)
SW공학 내 역할	초기 개발비용 절감	유지보수 효율화	복잡도 축소

생각 해보기

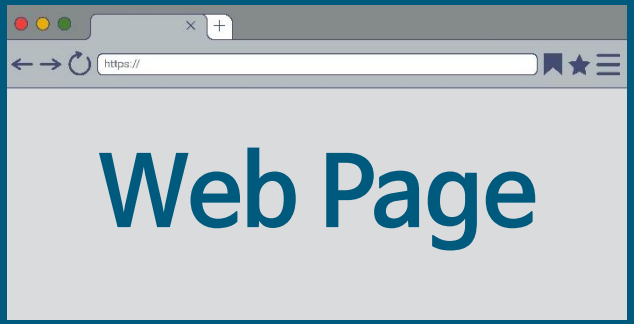
Q

객체지향 프로그래밍 방법론에서 클래스 기반 상속은 공통 기능을 재사용하고, 계층 구조를 표현하며, 다형성을 구현하기 위한 중요한 객체지향 기법입니다. 하지만 JavaScript는 전통적인 클래스 기반 상속이 아닌 프로토타입 기반 상속을 지원하고 있습니다.

프로토타입 기반의 상속이 기존의 클래스 상속과의 차이점은 무엇이며, JavaScript는 프로토타입 기반 언어인데, ES6 이후 class가 도입된 이유는 무엇일지 생각해 보고, 게시판에 올려 주세요.

01

DOM (Document Object Model)의 개념 이해



HTML + CSS + JavaScript

HTML의 모든 element를
객체로 해석

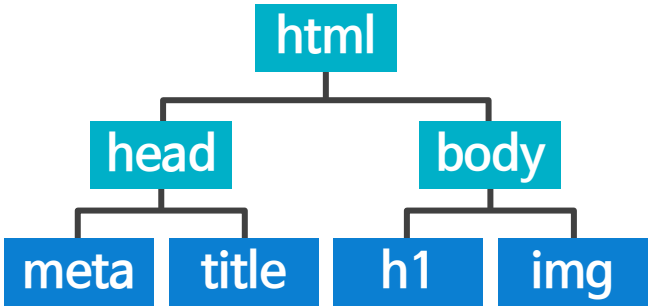
DOM 모델

1) 문서 객체 모델(DOM)이란

문서 객체 모델(DOM, Document Object Model)

웹 브라우저가 HTML 문서를 해석해서 만들어낸 트리 구조의 객체 모델 ➡ HTML 문서를 구조화(트리)한 설계도

```
<!DOCTYPE html>
<html lang="ko">
<head>
 <meta charset="UTF-8">
 <title>DOM Tree란?</title>
</head>
<body>
 <h1>DOM Tree</h1>
 <img src="images/test.jpg" alt="javascript">
</body>
</html>
```



02

DOM 트리 구조와 노드의 종류

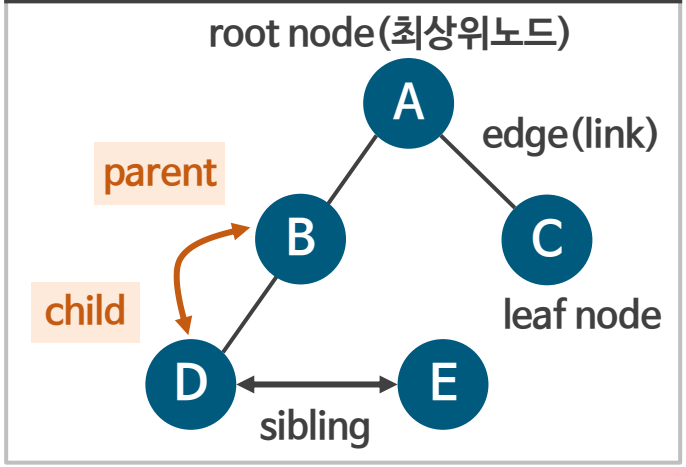
1) DOM 트리란?

"웹 문서에 있는 요소들 간의
부모, 자식 관계를 계층 구조로 표시한 것"

나무(Tree) 형태가 되기 때문에 "DOM 트리"라고 함

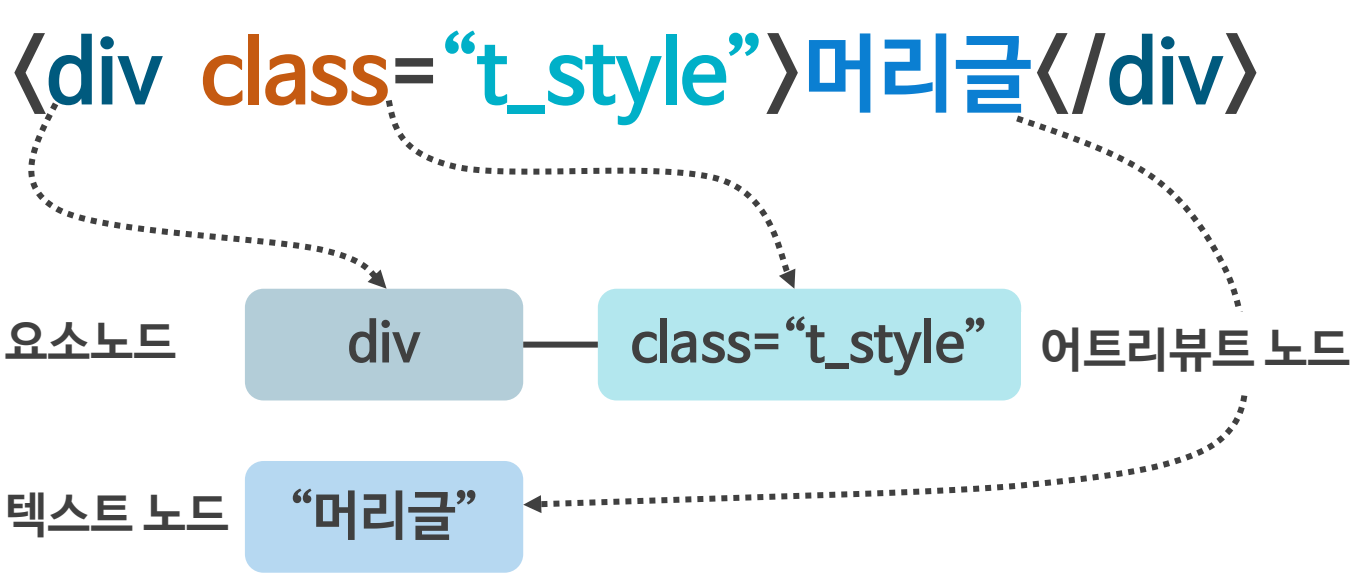
노드 (node)	문서의 구성 단위(요소, 속성, 텍스트 등)
엣지 (edge)	노드들 간의 부모-자식 관계를 연결한 선

트리 자료 구조



1) DOM 트리란?

세종사이버대학교



DOM 트리는 문서를 구조적으로 이해하고,
원하는 요소를 빠르게 찾고, 부분적으로 제어할 수 있게 기능 제공

2) DOM을 구성하는 원칙

문서 전체는 document 객체로 시작

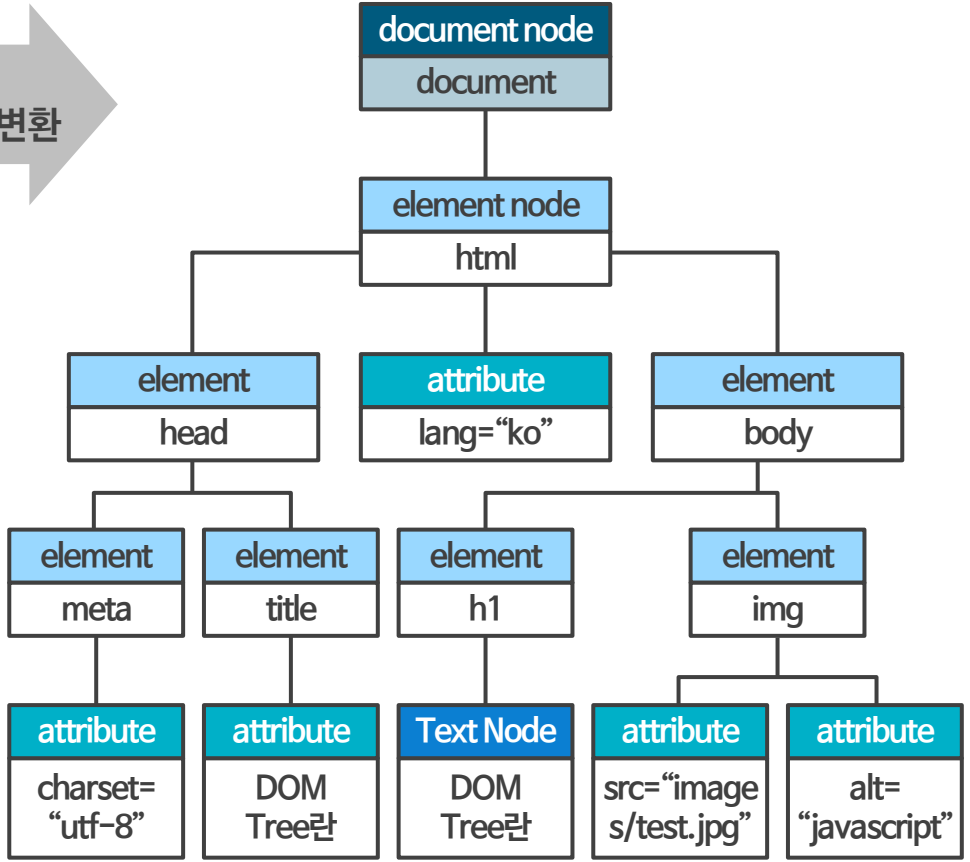
노드 객체 타입	설명
Element Node	HTML 요소(〈div〉, 〈p〉)
Text Node	요소 안의 텍스트
Attribute Node	속성(class, id)
Comment Node	주석(〈!-- --〉)

참고 브라우저 전체 실행 환경을 포함하면 최상위 객체는 window 객체가 됨

3) DOM 트리 예시


```
<!DOCTYPE html>
<html lang="ko">
<head>
 <meta charset="UTF-8">
 <title>DOM Tree란</title>
</head>
<body>
 <h1>DOM Tree</h1>
 <img src="images/test.jpg" alt="javascript">
</body>
</html>
```

DOM 트리로 변환

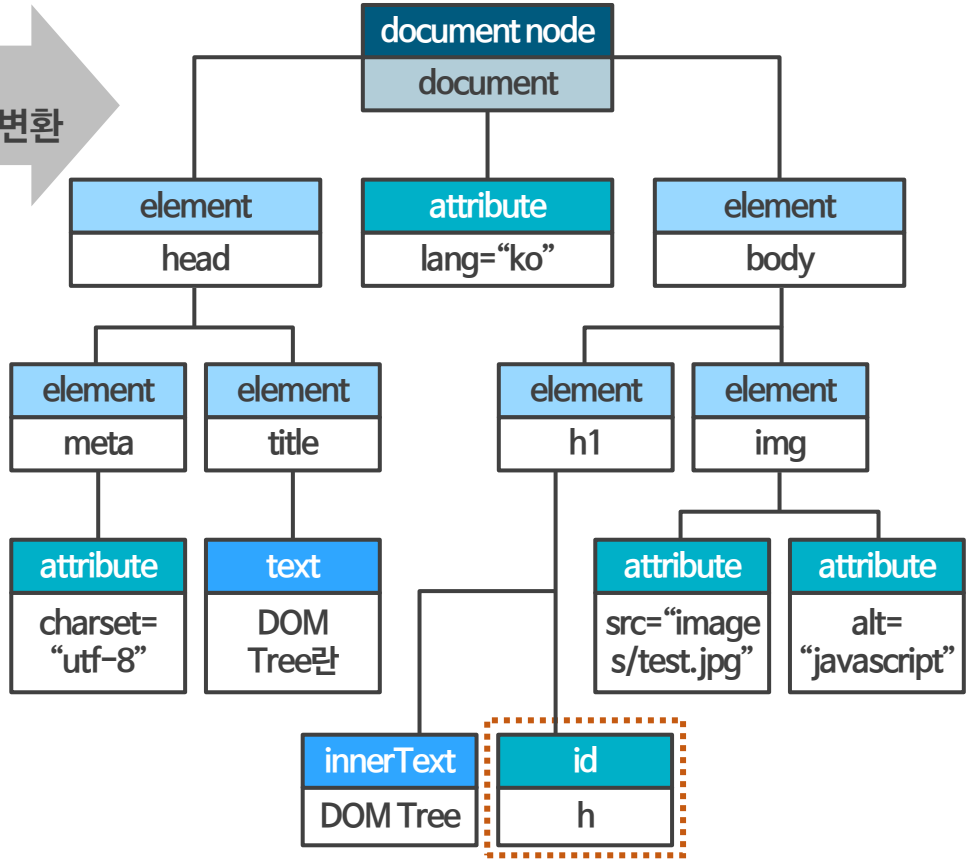


3) DOM 트리 예시

세종사이버대학교

```
<!DOCTYPE html>
<html lang="ko">
<head>
 <meta charset="UTF-8">
 <title>DOM Tree란</title>
</head>
<body>
 <h1 id="h">DOM Tree</h1>
 <img src="images/test.jpg" alt="javascript">
</body>
</html>
```

DOM 트리로 변환



03

DOM 요소 선택하기

1) DOM에서 요소 선택

모든 element는 document 객체의 하위 객체로 관리함

[기본형] getElement() 함수

```
document.getElementById("id명")
document.getElementsClassName("class명")
document.getElementsByTagName("태그명")
```

getElementsByClassName()과 getElementsByTagName()은 반환 값이 2개 이상일 수 있음

- HTMLCollection 객체에 저장됨

```
<h1 id="h">DOM Tree</h1>
//반환된 객체를 통해 HTML 요소 객체 제어
element = document.getElementById("h")
#id가 h인 요소를 반환 후, 태그 사이 Text를 변경
element.innerText = "Hello";
```

2) 자주 쓰는 선택자 함수

함수	반환값	설명
getElementById	단일 요소	• ID 기반
querySelector	단일 요소	• CSS 선택자 – 한 개의 값만 반환 – id 이름 앞에는 해시 기호(#), class 이름 앞에는 마침표(.), 태그는 기호 없이 태그명 사용
querySelectorAll	NodeList	• querySelecotorAll( ) 메소드는 반환 값이 여러 개일 때 모두 반환 ➡ 노드 리스트로 저장 • 여러 요소(forEach 가능)

04

DOM 요소 조작하기

1) DOM 트리에 노드 추가하기

[기본형] createElement() 함수

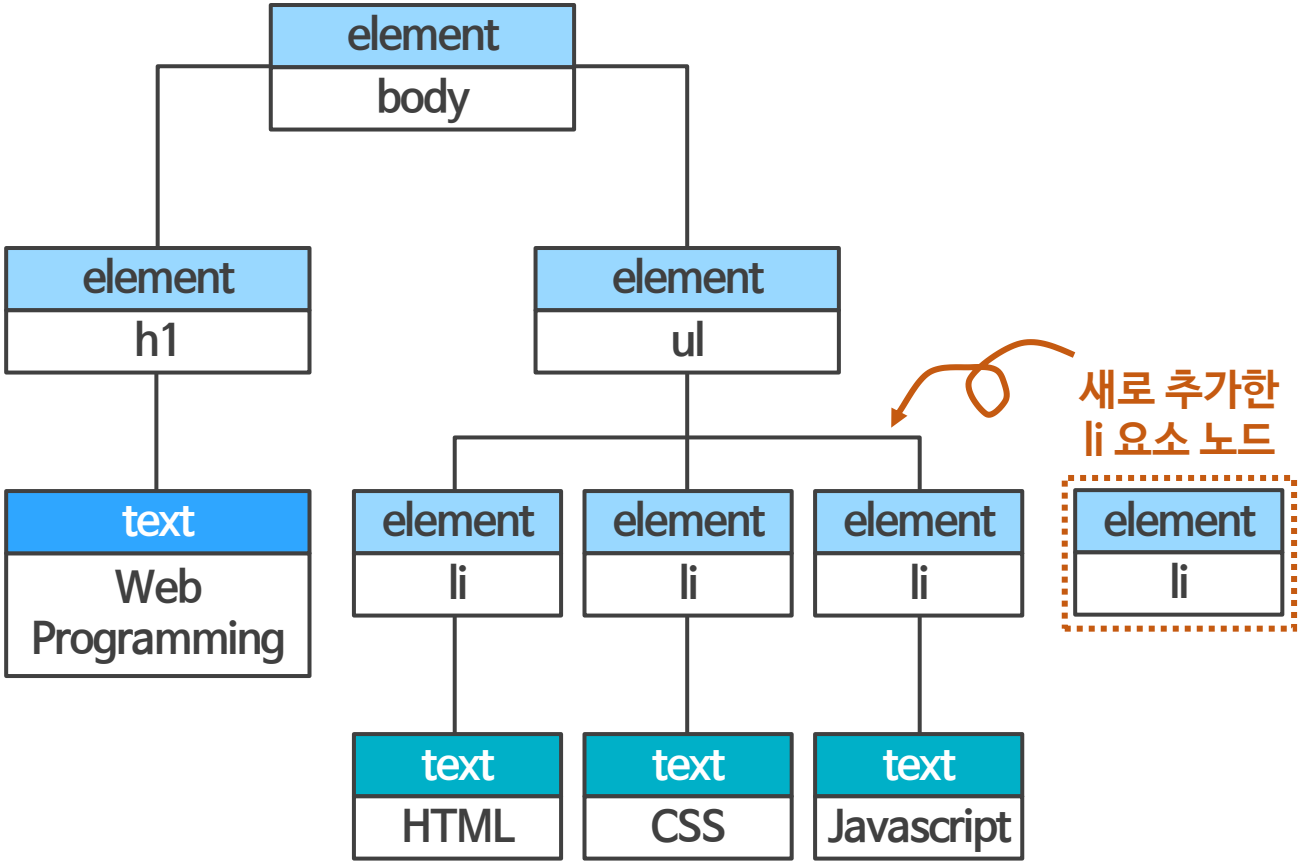
document.createElement("요소명") //계층구조에 맞춰 추가해야 함

```
<body>
<h1>Web Programming</h1>
<ul id="itemList">
  <li>HTML</li>
  <li>CSS</li>
  <li>Javascript</li>
</ul>
```

```
const ul = document.getElementById("itemList");
const newItem = document.createElement("li");

ul.appendChild(newItem);
```

1) DOM 트리에 노드 추가하기



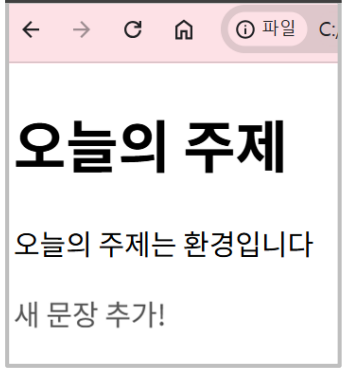
2) 예시


```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
 <body>
  <h1> 오늘의 주제 </h1>
  <p>오늘의 주제는 환경입니다 </p>

  <script>
                const p = document.createElement("p");
                p.textContent = "새 문장 추가!";
                document.body.appendChild(p);
  </script>

 </body>
</html>
```

실행 결과



2) 예시


```
<!DOCTYPE html>
…… 추가
<body>
 <h1> 오늘의 주세 </h1>
 <p>오늘의 주제는 환경입니다 </p>

 <script>
        const p = document.createElement("p");
        p.textContent = "새 문장 추가!";
        document.body.appendChild(p);

        p.setAttribute("class", "blue");
        p.style.color = "red";
        p.classList.add("active");

 </script>
</body>
</html>
```

실행 결과

세종사이버대학교

08주. 웹 기반 챗봇을 위한 UI 구현하기 : DOM과 이벤트 처리

02

이벤트(Event) 처리