Chapter 4 HTML과 JavaScript 연동하기

태그

JavaScript의 요소

자바스크립트 Core 문법

기본 문법과 구조, 데이터타입, 조건문, 반복문, 함수, 객체, 클래스(프로토타입)등이 포함

자바스크립트 Core 라이브러리

자바스크립트에서 기본으로 제공하는 문자열(String),Date, Math, Array, Number, 타이머함수를 비롯한 기타 전역함수등이 포함

자바스크립트 DOM (Document Object Model)

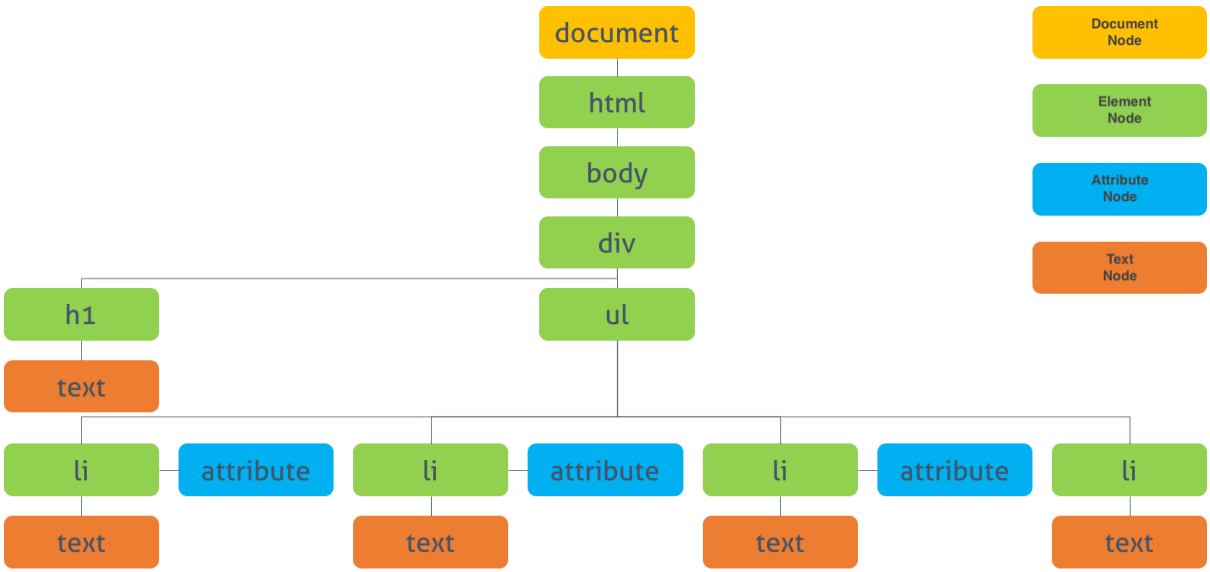
노드(node), 스타일, 속성, 이벤트, 위치 및 크기 등을 다룰 수 있는 다양한 기능이 포함

자바스크립트 BOM (Browser Object Model)

브라우저와 관련된 Window, Navigator, Location, History, Document, Screen 객체들이 포함

DOM (Document Object Model)

- DOM이란 문서객체모델이라고 하며 HTML 혹은 XML문서의 구조화된 표현을 제공하는 표준
- HTML에서는 자바스크립트가 DOM 구조에 접근할 수 있는 방법이 제공되며 이를 통해 문서 구조, 스타일, 내용등을 변경 할 수 있음
- DOM의 개체 구조는 **노드 트리(하나의 부모 줄기가 여러개의 자식 나뭇가지, 나뭇잎들을 가질 수 있는 나무와 같은 구조)**로 표현된다



DOM과 HTML의 차이점

DOM은 HTML 문서로부터 생성이 되지만 항상 동일하지 않다.

- **HTML** : 화면에 보이고자 하는 모양과 구조를 문서로 만든 것으로 단순 텍스트로 구성되어 있다. **(최초에 화면을 그릴때 사용하는 설계도)**
- **DOM** : HTML 문서의 내용과 구조가 객체 모델로 변화되어 다양한 프로그램에서 사용될 수 있다. **(설계도를 이용하여 실제로 화면에 나타내지는 인터페이스)**

HTML 요소 핸들링

자바스크립트에서 HTML DOM 요소에 접근하는 방법은 태그이름, 아이디, 클래스, 이름 등을 이용해 특정 노드 객체를 선택하는 것이다.

HTML 요소선택

> getElement(s)ByXXX() 를 이용해 특정 요소(태그) 노드를 가지고 옵니다.

```
document.getElementsByTagName("tag_name")
document.getElementById("id_name")
document.getElementsByClassName("class_name")
document.getElementsByName("name_attribute")
```

- 모든 `getElementsByXXX()` 함수의 경우 동일 조건의 모든 노드를 목록으로 가져옴.
- `getElementByName()`의 경우 태그의 name 속성으로 노드를 찾음.

```
let elist = document.getElementsByTagName("h2");

for(let el of elist) {
  console.log("##"+el.tagName);
}
```

쿼리 셀렉터

> getElementByXXX() 와 달리 단일 메서드로 원하는 요소를 찾을 수 있다. 단 유효한 CSS 셀렉터를 사용해야 하며 그렇지 않으면 예외가 발생한다.

```
document.querySelector("#main, #title, #footer");
document.querySelectorAll("p.note, p.tip")
```

- 콤마로 구분된 여러 셀렉터를 나열 할 수 있으며 해당 조건에 맞는 첫번째 노드만 가져옴.
- 해당 조건의 모든 노드를 가지고 오려면 querySelectorAll()을 사용 함.

HTML 요소 생성, 제거, 추가

> HTML 요소를 선택하는것 이외에도 특정 요소를 생성, 제거 하거나 자식 노드를 추가하는 것도 가능하다.

```
document.createElement(element)
document.removeChild(element)
document.appendChild(element)
document.replaceChild(element)
```

HTML 요소의 속성 핸들링

HTML 요소를 선택한 다음에는 해당 요소의 속성과 텍스트 노드에 접근할 수 있다. 물론 해당 요소내에 다른 요소들을 포함하고 있다면 마찬가지로 `getElement(s)XXX()` 함수를 사용해 자식노드들을 가지고 올 수 있다.

HTML 요소의 속성값 변경

> 해당 태그의 속성값을 변경할 수 있다. 이 경우 원래 HTML 소스가 바뀌는 것이 아니라 동적으로 내용만 변경되고 브라우저에 반영된다.

```
element.setAttribute(attribute, value)
element.getAttribute(attribute, value)
element.removeAttribute(attribute)
```

다음 예제는 `title-img` 라는 이름의 id 값으로 img 태그를 가져와 `src` 속성을 `b.jpg`로 변경하는 예 입니다. 이벤트와 연결해서 실행하면 동적으로 이미지가 바뀌게 된다.

```
const el = document.getElementById("title-img");
el.setAttribute('src','b.jpg');
```

HTML 이벤트핸들러 추가

> 해당 요소의 이벤트를 처리하는 방법으로 이벤트에서 살펴본것 처럼 이벤트 속성에 콜백 함수를 할당하는 형식으로 이벤트 핸들러를 추가할 수 있다.

```
`document.getElementById("id_name").onclick = **function**(){ code };`
```

`addEventHandler()`를 이용한 이벤트 핸들러 추가

> 이벤트 요소를 추가하기 위한 가장 구조적인 방법으로 onclick 속성에 콜백 함수를 구현하는 것 보다 좋은 방법이다.

```
document.getElementById("id_name").addEventHandler("click", function(){ code });
```

텍스트 노드 변경

> 해당 태그의 태그 바디에 있는 텍스트 노드의 값을 변경하기 위해서는 innerHTML 혹은 innerText 속성을 사용할 수 있다.

```
element.innerHTML = '<tag>text</tag>';
element.innerText = 'text';
```

- innerHTML 은 태그 바디에 다른 HTML태그와 자식노드를 포함한 텍스트 를 처리할 때 사용.
- innerText는 단순한 텍스트만을 처리할 때 사용.
- innerText는 CSS에 종속적으로 CSS에 의해 hidden 처리가 되어 있다면 텍스트노드를 읽을 수 없다.
- 따라서 가능하면 innerHTML사용을 권장.

<참고자료>

https://webclub.tistory.com/218

https://dinfree.com/lecture/frontend/123_js_3.html#02-domdocument-object-model

https://developer.mozilla.org/ko/docs/web/api/document_object_model/introduction

https://velog.io/@surim014/DOM이란-무엇인가#4-dom은-개발도구에서-보이는-것이-아니다