

HTML DOM(Document Object Model) Object

- ☞ BOM Object: Browser window에 대한 Object => BOM Object의 최상위 객체는 window Object
- ☞ DOM Object: Browser 안에 출력될 HTML window에 대한 Object => DOM Tree의 root는 document Object

☑ DOM Object의 구성요소

❖ HTML tag의 element

- ☞ tag name
- ☞ attribute
- ☞ css style
- ☞ event listener : HTML tag에 발생한 event
- ☞ content(innerHTML) : 시작 tag와 종료 tag 사이에 포함된 HTML content

위 5가지 정보들은 모두 DOM Object의 구성 요소가 된다.

❖ DOM Obejct의 구성 요소

- ☞ property : DOM Object의 member variable수. HTML tag의 attribute 반영
- ☞ method : DOM Object의 member function. HTML tag를 제어
- ☞ collection : 정보를 집합적으로 표현하는 일종의 array. 예를 들어 children collection은 DOM Object의 모든 child DOM Object에 대한 address를 가진다.
- ☞ eventListner : HTML tag에 작성된 event listener를 그대로 가진다.
- ☞ style : style property를 통해 HTML tag에 적용된 CSS style sheet에 접근 가능

❖ HTML tag와 DOM Object 사이의 관계

- ☞ DOM Object들은 DOM Tree에서 포함관계를 가지며 서로 parent, child, sibling의 관계로 연결되어 있다.
- ☞ 이들은 다음 4가지 property가 이용된다.
 - ✓ parentElement : parent object
 - ✓ children : 직계 child들의 collection
 - ✓ firstElementChild : 첫 번째 직계 child
 - ✓ lastElementChild : 마지막 직계 child

☞ 그리고 같은 sibling 관계에 있는 DOM Object 들은 다음 2개의 property로 접근한다.

- ✓ previousElementSibling
- ✓ nextElementSibling

☞ 모든 DOM Object 들은 위 6개의 property를 가지고 다른 DOM Object를 쉽게 접근할 수 있다.

아래는 예시 코드이다.

```
let div = document.getElementById("firstDiv");
let text = "div.id = " + div.id + "\n";

// id attribute가 "firstDiv"인 tag를 찾아 child element 갯수 찾기
text += "div.childElementCount = " + div.childElementCount + "\n";
```

☑ DOM 객체 다루기

❖ DOM Object 찾기 : document.getElementById()

☞ DOM Tree에서 특정 DOM Object를 찾아내고 싶을 때, id attribute value을 가지고 찾을 수 있다.

☞ 이 때 사용되는 것이 getElementById() method 이다. 이 method도 document Object의 member 이다.

아래 예문을 보자.

```
// id 값이 "testId"인 tag를 찾아 variable에 저장
let test = document.getElementById("testId");

// 해당 tag의 color attribute을 "red"로 변경
test.style.color = "red";
```

위의 코드처럼 어떤 object 들끼리 . 으로 연결해 variable = object.method(.getter()) 형식으로 가져 오는 것은 getter 이며,

variable name.property name(.하위 속성명) = "property value" 형식으로 쓰면 setter가 내부적으로 자동 실행된다.

또 다른 방법은 다음과 같이 cssText property에 style sheet를 직접 주어도 된다.

```
test.style.cssText = "background-color : gray"
```

❖ innerHTML property

- ☞ innerHTML property는 시작 tag와 종료 tag 사이에 들어있는 HTML content를 나타내는 string 정보이다.
- ☞ 이 tag를 이용하여 content를 바꿀수도 있다.

```
let test = document.getElementById("testId")
test.innerHTML = "바꾸고자 하는 내용의 텍스트"
```

그런데 단순 text 뿐만 아니라 또 다른 tag도 전달해 반영시킬 수 있다.

```
test.innerHTML = "바꾸고 싶은 <img src='cute.jpg'> text"
```

위와 같은 코드를 이용하면 text 사이에 image를 삽입한 content가 반영되게 된다.

❖ this

- ☞ Java와 같은 Object-Oriented Language에서는 익숙한 개념일 것이다.
- ☞ this는 object 자신을 가리키는 JavaScript keyword 이다.
- ☞ DOM Object에서 object 자신을 가리키는 용도로 사용된다.

```
<div onclick = "this.style.backgroundColor = 'red'">
```

위와 같은 코드가 있다고 했을 때, onclick은 Object를 click 하는 event가 발생했을 때를 뜻하며, click 시 '이 object의 / style property을 / 배경색=빨간색지' 으로 바꾸라는 뜻이 된다.