

JavaScript의 document Object

document는 DOM Tree의 최상위 Object 이다. Browser는 HTML document를 load 하기 전에 document object를 먼저 만든다. 그리고 document Object를 root로 하는 DOM Tree를 만든다.

☑ document Object의 역할

- ☞ property로 HTML document의 전반적 property를 나타낸다.
- ☞ method로 DOM Object를 검색한다.
- ☞ method로 새로운 DOM Object를 생성한다.
- ☞ method로 HTML document의 전반적 제어를 지원한다.

☑ Event Listener

- ☞ Event Listener는 70여개 이상의 event가 지원된다.
- ☞ 그 중에서 굉장히 많이 쓰이는 것들을 다음과 같다.
 - ✓ onclick
 - ✓ onerror
 - ✓ onmouseover
 - ✓ onscroll
 - ✓ onkeypress
 - ✓ onload
- ☞ 모두 on~ 이라는 keyword로 시작하는 것을 볼 수 있으며, '~한 event 발생 시' 라는 의미이다.
- ☞ 이 의미만 알면 그 뒤의 단어들은 직관적으로 naming 된다.

☑ document object의 주요 property, collection, method

❖ 주요 property

- ✓ title, body, head : htmldocument의 각 <title>, <body>, <head> tag
- ✓ URL : 현재 document의 URL
- ✓ location : 현재 document의 URL 정보를 가진 location object의 reference
- ✓ referrer : 이 HTML document를 load한 원래 document의 URL string. 이 document가 처음이면 빈 string

❖ 주요 collection

- ✓ images : document 내의 모든 object들의 collection
- ✓ links : document 내의 href attribute를 가진 모든 link object들의 collection (<a>나 <area>)
- ✓ forms : document 내의 모든 form object들의 collection

❖ 주요 method

- ✓ getElementById() : id name으로 첫 번째 DOM object를 return 한다.
- ✓ getElementsByTagName() : 특정 tag name을 가진 모든 tag collection 형태로 return 한다.
- ✓ getElementsByClassName() : 특정 class name을 가진 모든 tag collection 형태로 return 한다.
- ✓ open() : content를 모두 지우고 새로운 HTML content를 쓸 수 있도록 열기를 한다.
- ✓ write(), writeln() : document에 HTML content insert를 한다. ln이 붙은 것은 write() 후 '\n' 추가 출력을 한다. (Browser 상에는 빈칸 하나를 의미 한다.)
- ✓ close() : document object에 있는 HTML content를 Browser에 출력하고, 더 이상 write를 받지 않는다.

이 중에서 document.write()를 쓰게 되면 현재 document object에 담긴 내용의 끝에 HTML text를 추가하는 코드이다.

주의할 점은 HTML document가 load 되어 출력이 모두 이뤄진 후에는 document object가 '닫힌다'

이 상태에서 write()가 실행되면, Browser는 현재 document object에 담긴 document를 지우고 빈 document를 새로 연다.

현재 Browser Window에 출력된 HTML text를 지워버리고 새로운 내용을 출력하려면, document.open() -> document.write() -> document.close() 순으로 순서를 지켜 열고/닫기까지 확실히 해주어야 한다.

이 경우에도 document.close()가 실행된 후에는 HTML text를 덧붙일 수 없다.

그러나 아래와 같이 동적 구성을 통해 추가할 수 있다.

☑ HTML document의 동적 구성

HTML document의 load가 완료된 document에 새로운 HTML tag를 추가하는 방법은 추가하고 싶은 HTML tag를 나타내는 DOM object를 생성하여 DOM tree에 직접 삽입하는 것이다.

❖ DOM object 동적 생성

☞ `document.createElement("tag이름")` : HTML tag의 DOM object를 생성하는 코드

```
let newDIV = document.createElement("div");           // object 생성

newDIV.innerHTML = "새로 생성된 DIV 이다..";           // HTML text 삽입
newDIV.setAttribute("id", "newDiv");                   // attribute 추가
newDIV.style.backgroundColor = "red";                   // css style 추가
```

❖ DOM tree에 삽입

☞ DOM object를 DOM tree에 삽입하는 방법은 여러 가지가 있다. 그 중 다음 2가지 방법이 대표적이다.

```
☞ parent.appendChild(DOMobject);                       // 마지막 child로 삽입
☞ parent.insertBefore(DOMobject [, 기준자식]);         // child object 중 기준 child 앞에 삽입
```

❖ DOM object의 삭제

☞ DOM object의 삭제는 `removeChild()` method로 한다.

```
☞ parent.removeChild(child object);
```

예를 들어 `id="testId"` 인 DOM object를 DOM tree에서 삭제하고자 한다면 다음과 같이 한다.

```
let test = document.getElementById("testId");
let parent = testId.parentElement;

parent.removeChild(test);
```