**Web개발\_Day43\_노트정리**

**Text 객체**

ㆍdocument.getElementById로 해당 태그를 찾아 준 후, 태그 내부의 text를 받고 싶다면?

* 해당 태그를 찾아준 뒤, firstChild를 활용하여 내부 text를 가져올 수 있다.
* HTML영역에서 공백이 있다면, nextSibling으로 공백이 후의 태그에 접근할 수 있다.

ㆍText API

* Document객체에서 제공하는 내장함수의 개념, text에만 적용가능
* Text API의 종류
* appendData(“value값”) : 해당 문자열 value에 해당하는 문자열을 추가
* deleteData(start, end) : index중 (0번부터 채번) start부터 end만큼의 “길이”만큼을 삭제
* insertData(start, value) : (0번부터 채번) start부터 해당 value값을 삽입
* replaceData(start, end, value) : (0번부터 채번) start부터 end만큼의 “길이”만큼을 valule로 대체
* substringData(start, end) : (0번부터 채번) start부터 end만큼의 “길이”만큼의 값을 return

**이벤트**

ㆍevent란, 어떤 사건을 의미한다. 브라우저에서의 사건이란, 사용자가 클릭했을 ‘때’, 스크롤 했을 ‘때’, 필드의 내용을 바꾸었을 ‘때’와 같은 동작들을 의미한다.

ㆍonclick 이벤트 : 사용자가 이 버튼을 클릭 했을 ‘때’ 실행된다.

ㆍinline : 인라인 방식으로 이벤트를 등록하는 방식은 이벤트를 이벤트 대상의 태그속성으로 지정하는 것이다.

* <button onclick=” … “>

ㆍ프로퍼티 리스터

* 프로퍼티 리스너 방식은 이벤트 대상에 해당하는 객체의 프로퍼티로 이벤트를 등록하는 방식
* 인라인 방식에 비해서 HTML과 JavaScript를 분리할 수 있다는 점에서 선호되는 방식이다.
* Target.onclick = Function 이벤트명(event) { … }

ㆍevent객체 : 이벤트가 실행된 맥략의 정보가 필요할 대는 이벤트 객체를 사용한다.

ㆍaddEventListener

* addEventListener 방식은 이벤트를 등록하는 가장 권장되는 방식이다.
* 이 방식을 이용하면 여러 개의 이벤트 핸들러를 등록할 수 있다.
* Target.addEventListener( “click/submit/…등의 이벤트타입” function(event){ … });