■HTML특징 (웹문서를 만드는 언어)

웹문서의 표준
마크업언어
일반적인 텍스트 파일 (컴파일단계없이 브라우저에서 실행되는 언어)
컴퓨터 시스템과 운영체제에 독립적 (운영체제에 상관없이 브라우저만 있으면 실행됨)

■html5 추가된 태그 : 시멘틱 태그
의미있는 영역을 나타내는 태그가 추가됨
header
nav
section
aside
footer

■박스모델:
html태그 요소를 하나의 박스로 보고, 박스크기, 박스배경색, 여백 등 html 태그를 박스로 다루는 체계를 뜻한다.

박스모델의 종류

inline
inline-block
block

block : 한 줄 전체를 차지함, 크기를 가질 수 있다
: <p> , <h1>

inline : 크기를 가질 수 없다, 한줄내에 붙어서 나타남
 : 줄의 일부영역을 지정하여 스타일
<a>

inline-block : inline의 특성을 가지면서 크기를 가질 수 있다
<img>, <input>

■css 선택자(셀렉터)

공용선택자
태그선택자
자손선택자, 자식선택자 , 형제선택자
클래스 선택자
id 선택자
inlineStyle : 태그에 직접 style 지정

외부 css 가져오기 방법 2가지
1. html 문서 안에서 가져오기
<link href="" rel="stylesheet">

2. css 문서내에서 가져오기
@import url('https://fonts.googleapis.com/css2?family=Fredoka+One&family=Gaegu:wght@300&display=swap');

javascript

DOM : 문서내의 태그를 동적으로 제어하기 위해 제공되는 자바스크립트 객체 (Document Object Model)
Document객체로부터 DOM을 사용할 수 이다. (웹문서내의 있는 모든 태그들의 정보를 변경할 수 있다)

BOM: 브라우저를 제어하기 위해 제공되는 자바스크립트 객체 (Brower Object Model)
window 객체
window.open() , window.location.href="" , window.setInterVal()

■태그사이의 내용가져오기

innerHTML : html 태그 형식으로 그대로 가져옴
innterText : 사용자에게 '보여지는' 텍스트 값을 읽어옵니다.
textContent: 태그내의 모든 텍스트 값을 그대로 읽습니다.

<예제>

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>

<script>

		function getHTML() {
		    const element = document.getElementById('my_div');
		    alert(element.innerHTML);
		}

		function getTEXT() {
		   const element = document.getElementById('my_div');
		   alert(element.innerText);
		}

		function getCONTENT() {
		   const element = document.getElementById('my_div');
		   alert(element.textContent);
		}
</script>
</head>
<body>

<div id='my_div'>
    hello div~~~
  <span style='display:none'>사용자에게 보이지 않아요</span>
</div>

<input type='button'       value='innerHTML'          onclick='getHTML();'/>
<input type='button'       value='innerText'          onclick='getTEXT();'/>
<input type='button'       value='textContent'        onclick='getCONTENT();'/>
</body>
</html>
```

■이벤트

이벤트타겟 : 이벤트를 발생시킨 DOM 객체
이벤트객체: 발생한 이벤트에 관한 여러 정보를 가진 객체를 이벤트 객체
이벤트리스너 : 발생한 이벤트에 적절히 대처하기 위해 작성된 자바스크립트 코드를 말한다
(이벤트리스너: 이벤트감지하는 속성, 이벤트핸들러 : 이벤트가 감지되었을 때 수행되는 코드)

<예제>

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>

<script>
//이벤트 객체
// 이벤트를 처리하는 핸드러인 경우   핸드러함수 내에서 event 객체를 사용할 수 있음
// 발생한 이벤에 대한 정보를 객체로 만들어서 제공함


 function search(){
            // 키가 눌렀을때  이벤트 처리,  어떤 키를 눌렀는지 알 수 있음
            // alert( event.keyCode );
               console.log( event) ;

             if( event.keyCode ==13){
                        alert("검색중....");
             }
}

</script>
</head>
<body>


<input type="text" onKeyup="search();">
</body>
</html>
```

■이벤트 처리하기
이벤트 리스너 등록하기

1. 태그내에 직접등록하기 <button onclick="show();" >버튼 </button>
2. addEventListener("click" , 이벤트핸들러함수)
  : 자바스크립트 내에서 등록할 수 있다

3. 이벤트리스너속성 사용하기
   onclick= 이벤트핸들러함수

2,3의차이

: 2번은 버튼에 여러개의 click이벤트 리스너를 처리할 수 있다.
: 3번은 버튼에 여러개의 click이벤트를 등록할 경우 마지막에 등록된 것만 처리할 수 있다.

<예제 : 이벤트리스너 매서드를 이용하여 이벤트 등록하기
   addEventListener >

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>

<script>
    window.addEventListener("load" ,  init);

    function init(){
        let btn = document.getElementById("btn");
        btn.addEventListener("click" , function(){ alert("나 클릭1") ;});
        btn.addEventListener("click" , function(){ alert("나 클릭2") ;});
    }

</script>
</head>
<body>

<button  id="btn">버튼</button>
</body>
</html>
```

<예제 : 이벤트리스너 속성에 이벤트 등록하기>

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>

<script>
    window.addEventListener("load" ,  init)

    function init(){
        let btn = document.getElementById("btn");
        btn.onclick = function(){ alert("나 클릭1");}
        btn.onclick = function(){ alert("나 클릭2");}
    }
</script>
</head>
<body>


<button  id="btn">버튼</button>

</body>
</html>
```

HTML5 지오로케이션 API 사용하기

구글: 지오로케이션 API 사용하기 검색

Geolocation API는 사용자의 현재 위치를 가져오는 API로, 지도에 사용자 위치를 표시하는 등 다양한 용도로 사용할 수 있습니다.

지오로케이션이 사용자의 위치 결정하는 방법

1. GPS : 위성위치 확인 시스템
2. IP주소
3. 휴대폰
4. 와이파이

크롬: ip주소와 무선엑세스포인트의 주소가 전달 그 정보를 바탕으로 위치를 추적

Geolocation API 메소드

Method	설명
getCurrentPosition()	사용자의 현재 위치를 가져옴.
watchPosition()	사용자의 현재 위치를 가져오고 나서, 사용자의 움직임에 따라 지속적으로 위치 정보를 갱신함.
clearWatch()	watchPosition() 메소드의 실행을 중지함.

getCurrentPosition() 메소드의 반환값

속성	반환값
coords.latitude	**소수로 표현된 위도 값**
coords.longitude	**소수로 표현된 경도 값**
coords.accuracy	위도 값과 경도 값의 정확도
coords.altitude	평균 해수면을 기준으로 하는 고도 값(해발)
coords.altitudeAccuracy	고도 값의 정확도
coords.heading	북쪽을 기준으로 현재 진행 방향에 대한 시계방향으로의 각도 값(°)
coords.speed	초당 이동한 미터 수를 나타내는 속도 값(초속)
timestamp	위치 정보를 가져온 시간을 나타냄.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>

<button id = "find-me">Show my location</button><br/>
<p id = "status"></p>
<a id = "map-link" target="_blank"></a>


<script>

        function geoFindMe() {

            const status = document.querySelector('#status');
            const mapLink = document.querySelector('#map-link');

            mapLink.href = '';
            mapLink.textContent = '';

            // 성공했을 때 , 현재나의 위치 좌표를 전달받음
             function success(position) {
               const latitude  = position.coords.latitude;     //위도
               const longitude = position.coords.longitude;   //경도

               status.textContent = '';
                mapLink.href = `https://www.openstreetmap.org/#map=18/${latitude}/${longitude}`;
           //   mapLink.href = 'https://www.openstreetmap.org/#map=18/'+latitude + '/' + longitude;
                mapLink.textContent = `Latitude: ${latitude} °, Longitude: ${longitude} °`;
              }


             // 실패했을 때
             function error() {
               status.textContent = 'Unable to retrieve your location';
             }


           if(!navigator.geolocation) {
              status.textContent = 'Geolocation is not supported by your browser';
           } else {
              status.textContent = 'Locating...';
              navigator.geolocation.getCurrentPosition(success, error);
           }

        }

        //버튼에 클릭이벤트 등록하기
        document.querySelector('#find-me').addEventListener('click', geoFindMe);
</script>
</body>
</html>
```