

웹 프로그래밍 강의노트

11. Drag&Drop, 외부 웹 API 등

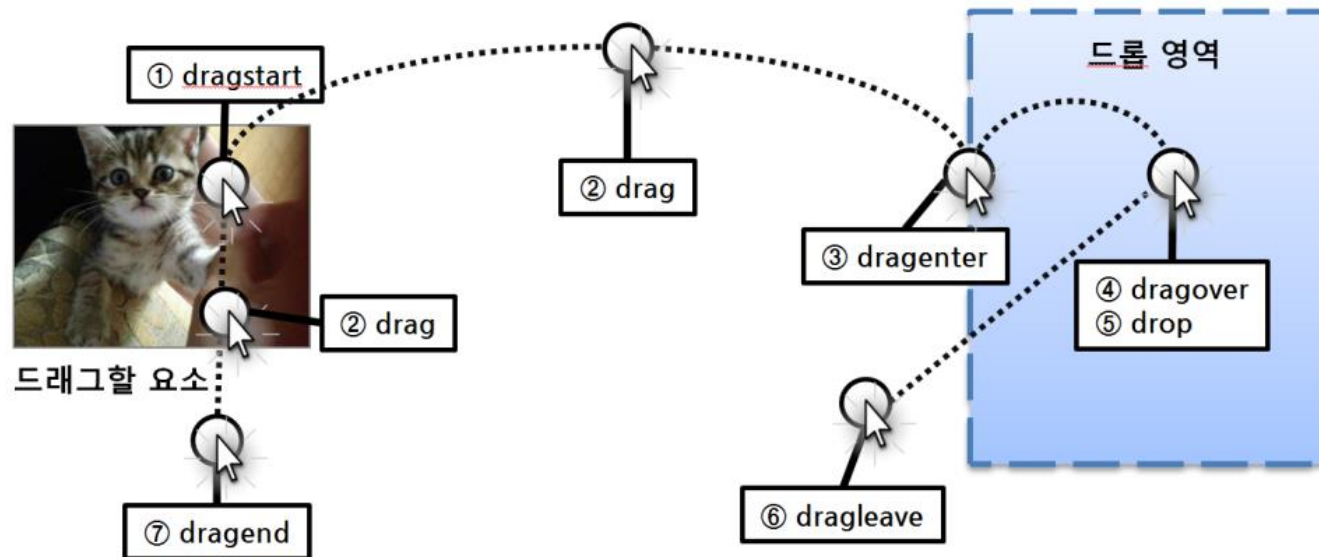
2023년 2학기

드래그 앤 드롭 사용하기

드래그 앤 드롭(drag & drop)

- **드래그 앤 드롭 이벤트**

- 화면에서 객체를 끌어다 놓음으로써 애플리케이션 간에 파일이나 데이터를 전달
- 끌기(drag) 과정과 놓기(drop) 과정



드래그 이벤트 사용하기

- **드래그 가능하도록 설정하기**
 - draggable 속성의 값이 true로 설정해야

```

```

- **드래그시 발생하는 이벤트**
 - dragstart : 드래그 시작 시 발생
 - drag : 드래그 도중 지속적으로 발생
 - dragend : 드롭 여부와 관계없이 드래그 종료시 발생
- **이벤트 핸들러 함수작성**

```
드래그할요소.이벤트이름 = function(e){  
    srcId = e.target.id;  
    // 처리하려는 작업  
}
```

예제: 드래그 정보 보여주기

```
<div id="dragsrc" style="display: block;">
  
  
  
</div>
<div id="dragprocess"></div>

<script type="text/javascript">
  var src = document.getElementById("dragsrc"),
  srcId, i=0;
  process = document.getElementById("dragprocess");

  src.ondragstart = function(e){
    srcId = e.target.id;
    process.innerHTML = srcId + "의 드래그가 시작되었습니다.<br/>";
  }
  src.ondrag = function(e){
    process.innerHTML += ".";
  }
  src.ondragend = function(e){
    process.innerHTML = srcId + "의 드래그가 끝났습니다.<br/>";
  }
</script>
```



apple의 드래그가 끝났습니다.

드롭 이벤트 사용하기

- 드롭(drop) 영역에 관련된 이벤트
 - dragenter : 드롭 영역에 진입할 때
 - dragover : 드롭 영역 위에 있을 때
 - dragleave : 드롭 영역에서 벗어날 때
 - drop : 드래그 요소를 드롭 영역에 놓을 때
- 드롭 이벤트 허용 : preventDefault() 메소드
 - 드롭영역 요소에 이미 정의되어 있는 이벤트를 중지하고 현재 이벤트가 발생되도록

```
target.ondragover = function(e) {  
    e.preventDefault();  
    if(isDraggingOver == false) isDraggingOver = true;  
}
```

- 데이터 전달 : dataTransfer 객체
 - 드롭 이벤트 발생시에 데이터를 처리하도록
 - dataTransfer 객체의 setData() 메소드에서 저장하고, getData() 메소드에서 지정 형식의 데이터를 꺼내온다

```
src.ondragstart = function(e) {  
    e.dataTransfer.setData("text", e.target.id);  
}  
  
target.ondrop = function(e) {  
    var data = e.dataTransfer.getData("text");  
    e.target.appendChild(document.getElementById(data));  
}
```

```

<div id="dragsrc" style="display: block;">
  
  
  
</div>

<div id="droptarget"></div>
<div id="dropprocess"></div>

<script type="text/javascript">
  var droptarget = document.getElementById("droptarget");
  var dropmessage = document.getElementById("dropprocess");
  dropmessage.innerHTML = "message";

  var src = document.getElementById("dragsrc");
  var srcId;

  src.ondragstart = function(e){ srcId = e.target.id; }

  droptarget.ondragover = function(e) {
    e.preventDefault();
    dropmessage.innerHTML = "과일이 배달되었네";
  }

  droptarget.ondragleave = function(e) {
    e.preventDefault();
    dropmessage.innerHTML = "과일이 떠났어요";
  }

  droptarget.ondrop = function(e) {
    e.preventDefault();
    dropmessage.innerHTML = "냠냠 맛있는 " + srcId;
  }

</script>

```


위치정보 사용하기

지오로케이션 API의 개요

단발성 위치 요청하기

반복적 위치 요청하기

위치정보(geolocation)

- 위치정보
 - GPS가 내장된 모바일 기기에서 정확한 위치 파악 가능
 - 위치 정보(위도, 경도)를 조회하여 어플리케이션에 활용
 - 사용예
 - 사용자에게 길안내, 혹은 근처 편의시설 정보 제공
 - 주변 교통 정보 등을 활용한 다양한 웹서비스
 - 지도와 결합하여 자동차 내비게이션과 같은 서비스
- HTML5 표준 API
 - 위치정보를 얻을 수 있는 자바스크립트 API 제공

지오로케이션 API

- 내장객체인 navigator.geolocation 을 이용
 - 단발성 위치 요청
 - getCurrentPosition() : 현재 위치 정보를 반환
 - 반복적 위치 요청
 - watchPosition() : 지속적으로 갱신되는 위치정보
 - clearWatch() : 추적 취소
- 브라우저에서 지오로케이션 API 지원 여부 확인

```
if (navigator.geolocation) {  
    // 지오로케이션 사용 가능. 원하는 작업 처리하는 프로그램  
} else {  
    // 지오로케이션 기능 지원하지 않음. 에러처리 프로그램  
}
```

단발성 위치 요청하기

- 현재 위치정보 구하기

- geolocation 객체의 `getCurrentPosition()` 메소드
 - `getCurrentPosition(성공콜백함수, 에러콜백함수, 옵션)`
 - 위치정보 요청이 성공하면 성공콜백함수를 실행
- 콜백함수의 인자인 `Position` 객체에 위치정보 전달
 - ```
function myCallback(myPos) {
 // Position 객체인 myPos에 전달된 위치정보 사용, 작업수행
}
```

- `Position` 객체

- `coords.latitude` 위도 , `coords.longitude` 경도
- `coords.altitude` 고도 , `timestamp` 시각

```

<head>
 <script type="text/javascript">
 function getLocation() {
 if (navigator.geolocation) {
 navigator.geolocation.getCurrentPosition(myCallback);
 } else {
 alert("지오로케이션이 지원되지 않습니다.");
 }
 }
 function myCallback(myPosition) {
 var latitude = myPosition.coords.latitude,
 longitude = myPosition.coords.longitude,
 date = new Date(myPosition.timestamp),
 parsedDate = date.toUTCString();

 document.getElementById("display").innerHTML = "위치: 일시 " + parsedDate
 + "에 위도 " + latitude + ", 경도 " + longitude + "에 있습니다. "
 }
 </script>
</head>
<body>
 <h3>단발성 위치정보 요청하기</h3>
 <button onclick="getLocation();" >위치정보 확인</button>
 <div id="display">위치: </div>
</body>

```

# 반복적 위치 요청하기

- 변경된 위치정보 가져오기
  - 내비게이터와 같이 사용자 위치를 추적하거나 갱신된 위치정보를 가져와야 할 때
  - 주기적이 아니라 변경된 경우에만 위치정보 가져온다
  - geolocation의 watchPosition(), clearWatch() 메소드 사용

- 위치 추적: watchPosition() 메소드

- getCurrentPosition() 메소드의 인자와 의미가 같음
- 위치정보 바뀔 때마다 watchPosition()의 콜백함수 실행

```
var watchID = navigator.geolocation.watchPosition(myCallback2);
function myCallback2(myPos) {
 // Position 객체인 myPos에 전달받은 위치정보를 사용하여 원하는 작업 수행
};
```

- 추적 중지: clearWatch() 메소드

- watchPosition() 메소드가 반환한 ID값을 clearWatch() 의 인자로 사용하여 전달

```
navigator.geolocation.clearWatch(watchID);
```

```

<head>
 <script>
 var watchID, i=0;

 function startMyLocation() {
 if (navigator.geolocation) {
 watchID = navigator.geolocation.watchPosition(myCallback);
 } else {
 alert("지오로케이션이 지원되지 않습니다."); }
 }
 function myCallback(myPosition) {
 var latitude = myPosition.coords.latitude,
 longitude = myPosition.coords.longitude,
 date = new Date(myPosition.timestamp),
 parsedDate = date.toUTCString();
 document.getElementById("display").innerHTML += '
 ['+(i++)+'] 일시 '+parsedDate
 +', 위도 ' + latitude + ', 경도 ' + longitude ;
 }
 function stopMyLocation(){
 if (navigator.geolocation){
 navigator.geolocation.clearWatch(watchID);
 document.getElementById("display").innerHTML += '
 ** 추적 종료! **' ;
 }else{
 alert("지오로케이션이 지원되지 않습니다."); }
 }
 </script>
</head>
<body>
 <h3>위치정보 추적 요청하기</h3>
 <button onclick="startMyLocation();">위치정보 추적 시작</button>
 <button onclick="stopMyLocation();">위치정보 추적 종료</button>
 <div id="display">위치: </div>
</body>

```



# 외부 웹 API 사용하기

제3자 API의 특징

카카오 지도 웹 API 사용하기

카카오 지도 API의 추가 기능

# 제3자 API의 특징

- 웹 API , 제3자(Third Party) API
  - 외부 기업/기관이 자신의 기능을 Open API로 제공
    - 해당 기능 손쉽게 구현 가능
  - 구글, 네이버, 카카오 등 웹 서비스 기업 혹은 공공기관
    - 지도, 단어사전, 소셜미디어 등을 Open API로 매우 쉽게 구현
    - 페이스북, 트위터 등 기존의 소셜미디어와 연동

# 제3자 API의 예와 특징

- 대표적인 제3자 API
  - 구글 개발자 사이트 (<https://developers.google.com>)
    - Map, Search, TensorFlow, Cloud, YouTube 등 다양한 기능의 API
  - 네이버 개발자 센터(<https://developers.naver.com/docs>) :
    - 검색, 지도, 로그인, 음성인식 및 음성합성, 기계번역 등 API
  - 카카오 개발자 센터(<https://developers.kakao.com>) :
    - 로그인, 카카오톡, 지도, 비전 등 다양한 기능을 API로 제공
  - 소셜미디어 사이트 : 페이스북, 트위터, 인스타그램 등
    - 소셜미디어 사이트에 연결 혹은 기능을 활용하는 API를 제공
- Open API의 경우 대부분 무료
  - 경우에 따라서 키를 발급하여 API 사용량 제어

# 카카오 지도 웹 API 사용하기

- 참조: “카카오 지도 웹 API 가이드”
  - 카카오 개발자 센터에서 제공
  - <https://apis.map.kakao.com/web/guide/>
- 카카오 지도 웹 API 사용을 위한 appkey 발급 절차
  - 1) 카카오 개발자 사이트 접속 (<https://developers.kakao.com>)
  - 2) 개발자 등록을 하고 앱을 생성
  - 3) 플랫폼을 등록. 웹에서 사용하려면 “웹 플랫폼 등록” 선택
  - 4) 웹사이트 도메인을 등록. (예, <http://localhost:8080>)  
등록된 도메인에서 지도 API를 사용 가능
  - 5) 앱 키 항목에서 “Javascript 키”를 선택하여 지도 API에서 appkey의 값으로 사용



앱 키

|              |                   |
|--------------|-------------------|
| 네이티브 앱 키     | be[redacted]4aa   |
| REST API 키   | d6[redacted]d3c8  |
| JavaScript 키 | 08[redacted]936   |
| Admin 키      | 298[redacted]b291 |

Web

삭제 수정

|         |                                                |
|---------|------------------------------------------------|
| 사이트 도메인 | https://dknife.github.io<br>https://github.com |
|---------|------------------------------------------------|

• 카카오 로그인 사용 시 Redirect URI를 등록해야 합니다. [등록하러 가기](#)

# 지도 그리는 과정

- 지도 영역 지정 및 지도 API 호출

- 웹페이지 내에 지도를 그릴 영역 <div> 지정
- 발급받은 키로 카카오 지도 API 호출

```
<div id="map" style="width:600px; height:300px;"></div>
<script src="//dapi.kakao.com/v2/maps/sdk.js?appkey=...발급받은 키값..." />
```

- 지도 그리기

- 지도 영역 요소를 id 로 찾아와서 변수로 저장
- 지도에 적용할 옵션 사항들을 설정하여 변수로 저장
- 이들 변수를 지도그리기 객체 kakao.maps.Map에 전달

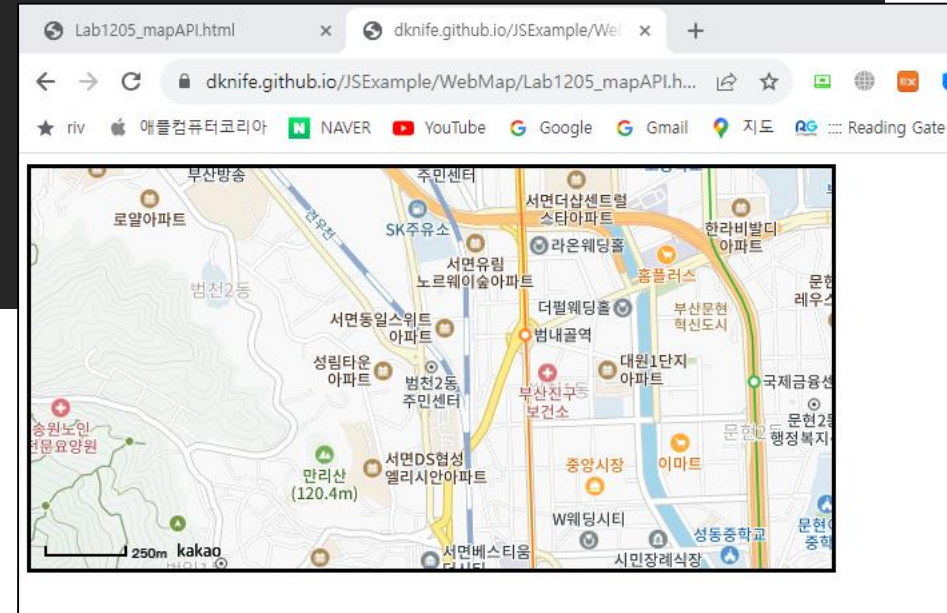
```
var mapContainer = document.getElementById('map') // 지도 그릴 영역
var mapOption = { center: .. , level : ... , ... } // 지도 그리기 옵션
var map=new kakao.maps.Map(mapContainer, mapOption); // 지도 생성하기
```

```

<body>
<div id="map" style="border: solid; width:600px; height:300px;"></div>
<script type="text/javascript"
src="//dapi.kakao.com/v2/maps/sdk.js?appkey=08*****936"></script>

<script type="text/javascript">
 var mapContainer = document.getElementById('map'), // 지도를 표시할 div
 mapOption = {
 center: new kakao.maps.LatLng(35.146497, 129.056171), // 지도의 중심좌표
 level: 5, // 지도의 확대 레벨
 };
 // 지도를 생성한다
 var map=new kakao.maps.Map(mapContainer, mapOption);
</script>
</body>

```



# 카카오 지도 API의 추가 기능

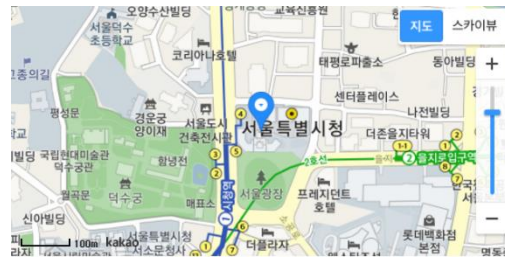
- 컨트롤 위젯과 마커 추가하기

- 지도 타입 변경 컨트롤

- MapTypeControl() 메소드로 컨트롤 객체 생성
    - addControl() 메소드로 컨트롤 추가, 위치는 ControlPosition 속성

```
var mapTypeControl = new kakao.maps.MapTypeControl(); // 컨트롤 생성
map.addControl(mapTypeControl, kakao.maps.ControlPosition.TOPRIGHT);
```

- 지도 타입 변경 컨트롤에서 “일반지도”와 “스카이뷰” 중 선택



일반지도



스카이뷰



- 확대 축소 컨트롤

- ZoomControl() 메소드로 객체 생성, addControl()로 컨트롤 추가

```
var zoomControl = new kakao.maps.ZoomControl(); // 지도 확대축소 컨트롤 생성
map.addControl(zoomControl, kakao.maps.ControlPosition.RIGHT); // 컨트롤 추가
```

- 다양한 기능을 라이브러리로 제공

- Marker()에서 마커 객체의 위치와 드래깅 가능 여부 등 설정

```
var marker = new kakao.maps.Marker({ // 지도에 마커 생성
 position: ... , draggable: ..., map: ... // 마커 위치, 드래깅 가능 여부, 지도 객체
});
```

마커



확대 축소 레벨 6



확대 축소 레벨 8

- 이벤트 처리 기능 추가하기

- 카카오 지도 API에서 처리 가능한 이벤트

- 중심좌표 변화('center\_changed'), 확대축소('zoom\_changed'), 영역 변화('bounds\_changed'), 클릭('click'), 드래그('drag') 등

- 이벤트 처리기

- 지도의 event 객체 addListener(지도, 이벤트, 콜백함수) 메소드



클릭 이벤트를 처리한 결과

(중간 생략)

```
var map = new kakao.maps.Map(mapContainer, mapOption); // 지도 생성
```

```
var mapTypeControl = new kakao.maps.MapTypeControl(); // 지도 타입 변경 컨트롤 생성
map.addControl(mapTypeControl, kakao.maps.ControlPosition.TOPRIGHT);
```

```
var zoomControl = new kakao.maps.ZoomControl(); // 지도에 확대 축소 컨트롤 생성
map.addControl(zoomControl, kakao.maps.ControlPosition.RIGHT);
```

```
var marker = new kakao.maps.Marker({ // 지도에 마커 생성
 position: new kakao.maps.LatLng(37.566535, 126.977969), // 마커의 좌표
 draggable: true, // 마커를 드래그 가능하도록 설정
 map: map // 마커를 표시할 지도 객체
});
kakao.maps.event.addListener(map, 'click', function (mouseEvent) { // 지도 클릭 이벤트 등록
 console.log('지도에서 클릭한 위치의 좌표는 ' + mouseEvent.latLng.toString() + ' 입니다.');
```

```
 alert('지도에서 클릭한 위치의 좌표는 ' + mouseEvent.latLng.toString() + ' 입니다.');
```

```
});
```

