

## FingerEyes-Xr for HTML5 Tutorials - 04

# 속성 확인(Identify) 기능

2015년 3월 2일, 1차 배포





## 속성 확인(Identify) 기능 구현하기

여기서 말하는 속성(Attribute)은 공간 데이터와 연관된 데이터를 말합니다. GIS에서 Identify 기능은 GIS 분야의 표준 용어는 아니지만 통상 Identify 기능이라 함은 화면에 표시된 도형을 마우스로 클릭했을 때 클릭된 도형과 연관된 속성 정보를 사용자에게 제공하는 것입니다. 공간 데이터로써의 도형과 속성은 수치지도 레이어인 ShapeMapLayer와 WFSLayer를 통해 제공됩니다. 이 장은 ShapeMapLayer 레이어를 이용한 Identify 기능에 대해 설명합니다. WFSLayer의 경우에도 ShapeMapLayer 레이어와 동일한 방식으로 Identify 기능을 구현하여 사용자에게 제공할 수 있습니다.

FingerEyes-Xr은 Flex 버전과 HTML5 버전이 존재하며 이 글은 HTML5에 대해 글입니다. FingerEyes-Xr for HTML5에 대한 소스 코드는 GitHub에서 다운로드 받을 수 있으며 URL은 <https://github.com/FingerEyes-Xr/src> 입니다. 아래의 화면은 FingerEyes-Xr for HTML5에 대한 GitHub 웹 화면입니다.

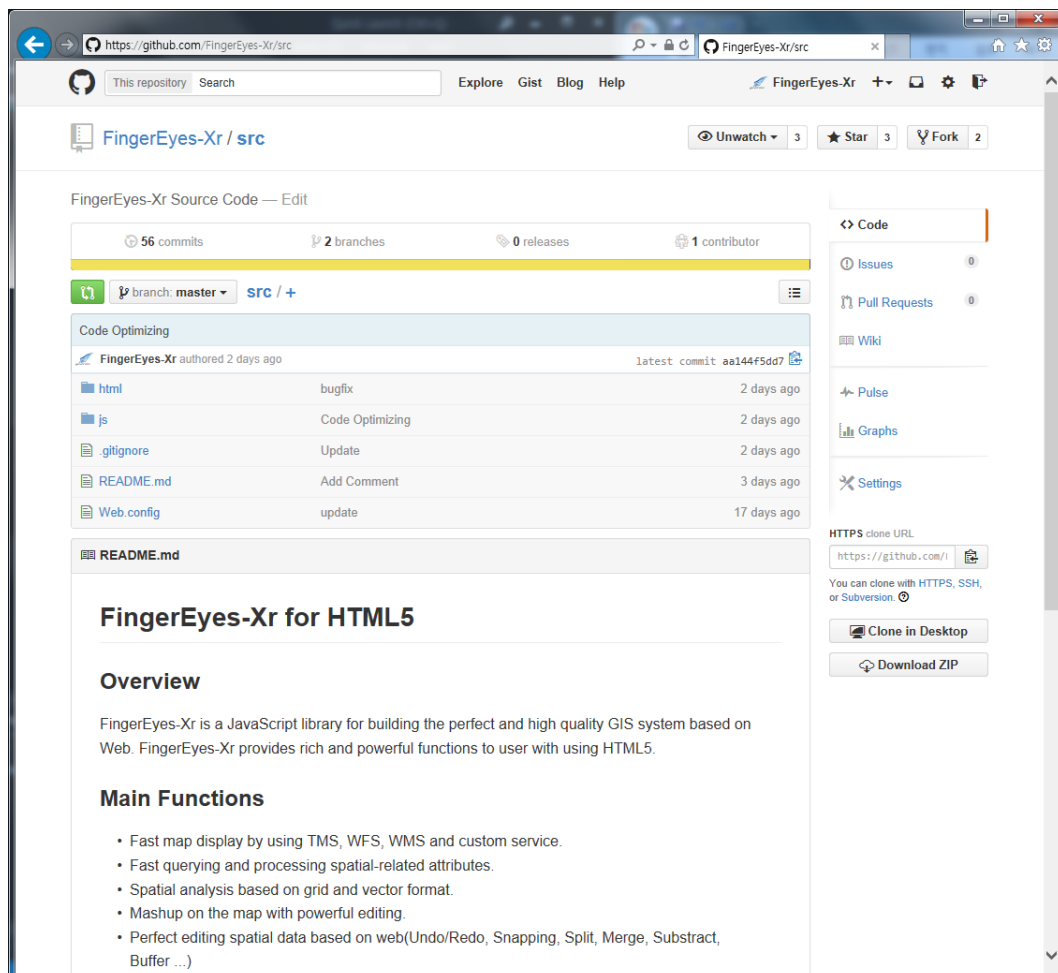


그림 1. FingerEyes-Xr for HTML5에 대한 GitHub

필자는 이 글에 대한 예제 코드를 VisualStudio 2012를 이용하여 작성 하였습니다. 독자 여러분들이 어떤 툴을 사용하든 문제는 없겠으나 VisualStudio 2012에서 제공하는 JavaScript 디버깅과 기본적으로 제공하는 웹서버(IIS)를 통해 웹 기반의 프로그래밍을 편리하고 효율적으로 진행할 수 있었기에 선택하였습니다.

참고로 웹 기반의 프로그래밍은 웹서버를 이용하여 URL 형태로 접근하지 않으면 올바르게 실행되지 않으므로 반드시 웹서버를 통해 URL 형태로 접근하시기 바랍니다.

이제 Identify 기능에 대한 코드를 작성해 보겠습니다. 먼저 map.html 파일을 웹서버에서 접근할 수 있는 곳에 생성합니다. 필요하다면 map.html이 아닌 다른 파일명으로 생성 하여도 진행에 문제는 없습니다. 다만, URL을 통해 웹브라우저에서 실행할 수 있는 경로에 생성한다는 것이 중요합니다. 일반 html 파일을 생성하고 다음처럼 입력합니다.

```
01 <html xmlns="http://www.w3.org/1999/xhtml">
02 <head>
03   <title>FingerEyes-Xr : Identify</title>
04 </head>
05 <body>
06
07 </body>
08 </html>
```

**코드 1. 초기 map.html**

위의 코드에서 3번 라인 밑에 다음 코드를 추가합니다.

```
01 <script src="http://www.geoservice.co.kr/z/Xr.min.1.0.js"></script>
02
03 <script type="text/javascript">
04
05 </script>
```

**코드 2. 기본 Script**

1번 코드에서 CDN(Content Delivery Network)을 사용하여 FingerEyes-Xr의 JavaScript 라이브러리에 대한 소스 코드를 추가 하고 있습니다. 그리고 4번의 빈 공백 부분에 앞으로 추가할 JavaScript 코드가 입력될 것입니다.

UI를 구성하기 위해 <body> 부분을 다음처럼 변경합니다.

```
01 <body>
02   <div id="mainLayout">
03     <div id="mapDiv"></div>
04     <div id="title">
05       FingerEyes-Xr for HTML5 :
06       <font color="#349bd6">Identify Example</font>
07       <br />
08       (click the building ...)
09     </div>
10   </div>
11 </body>
```

**코드 3. 기본 UI 구성**

특히 Id가 mapDiv인 DIV 요소에 맵(Map)이 표시될 것입니다. 일단 여기까지 작성하고 실행해 보면 다음과 같은 화면이 나타나는 것을 볼 수 있습니다.

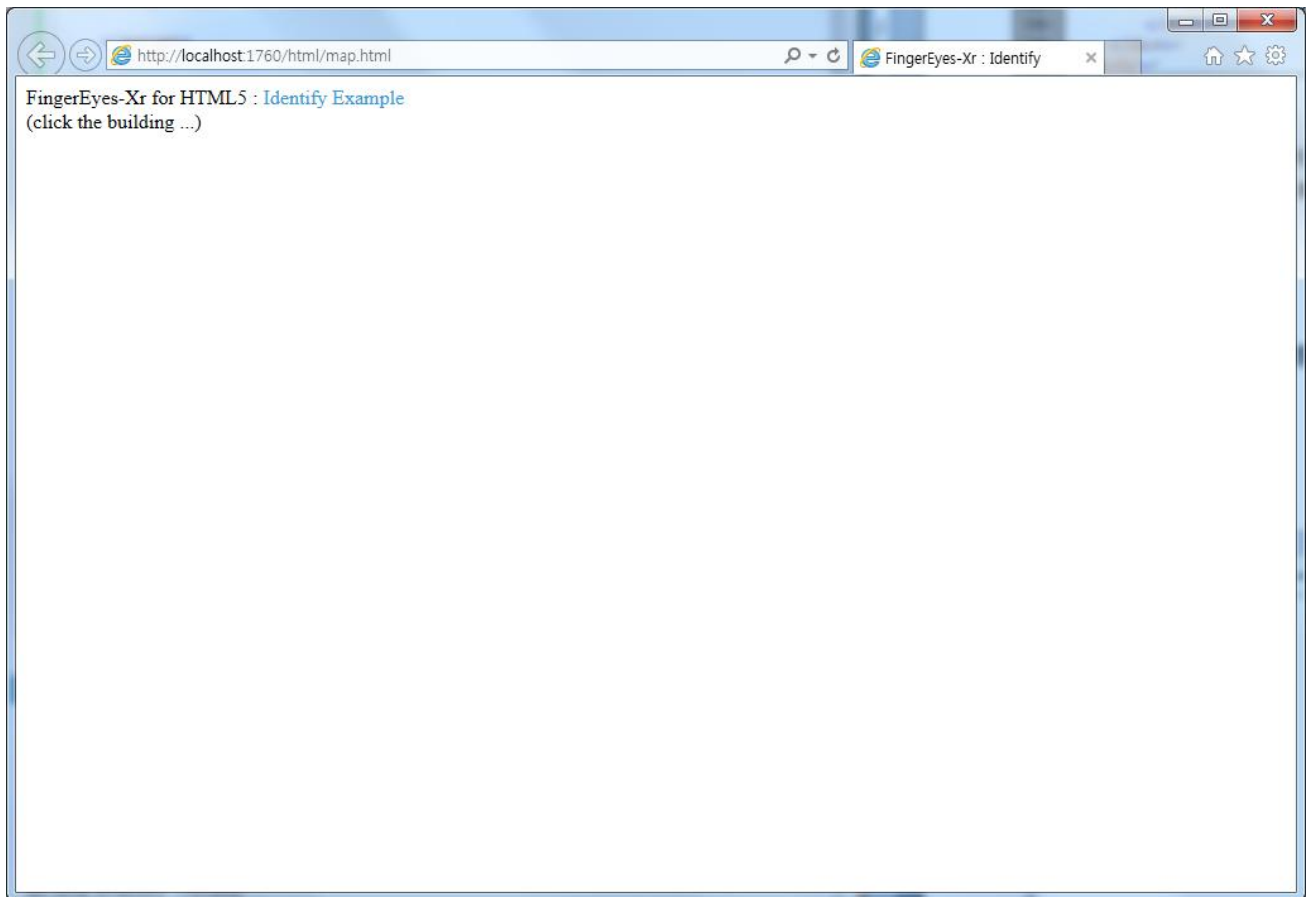


그림 2. 기본 UI 구성에 대한 실행 화면

특별히 어떤 버튼 등과 같은 UI를 제공하지 않습니다. 건물에 대한 수치지도를 표시할 것이고 마우스로 건물을 클릭하면 클릭된 도형의 속성을 화면에 표시할 것입니다.

이제 만든 페이지에 스타일을 적용해 보도록 하겠습니다. <head> 바로 밑에 아래의 스타일 코드를 추가합니다.

```

01 <style>
02   body
03   {
04     margin:0px;
05     padding:0px;
06   }
07
08   #mainLayout
09   {
10     position:relative;
11     width:100%;
12     height:100%;
13     border:none;
14   }
15
16   #mapDiv {
17     top:0px;
18     left:0px;
19     position:absolute;
20     width:100%;

```

```

21     height:100%;
22     border:none;
23     overflow:auto;
24 }
25
26 #title {
27     top:12px;
28     left:12px;
29     padding: 12px;
30     position:absolute;
31     background:rgba(0,0,0,0.7);
32     border:none;
33     overflow:auto;
34     border-radius: 12px;
35     font-size: 24px;
36     color: #ffffff;
37     font-family: "Arial";
38 }
39 </style>

```

코드 4. UI에 대한 스타일 적용

스타일이 적용된 페이지는 다음과 같습니다.

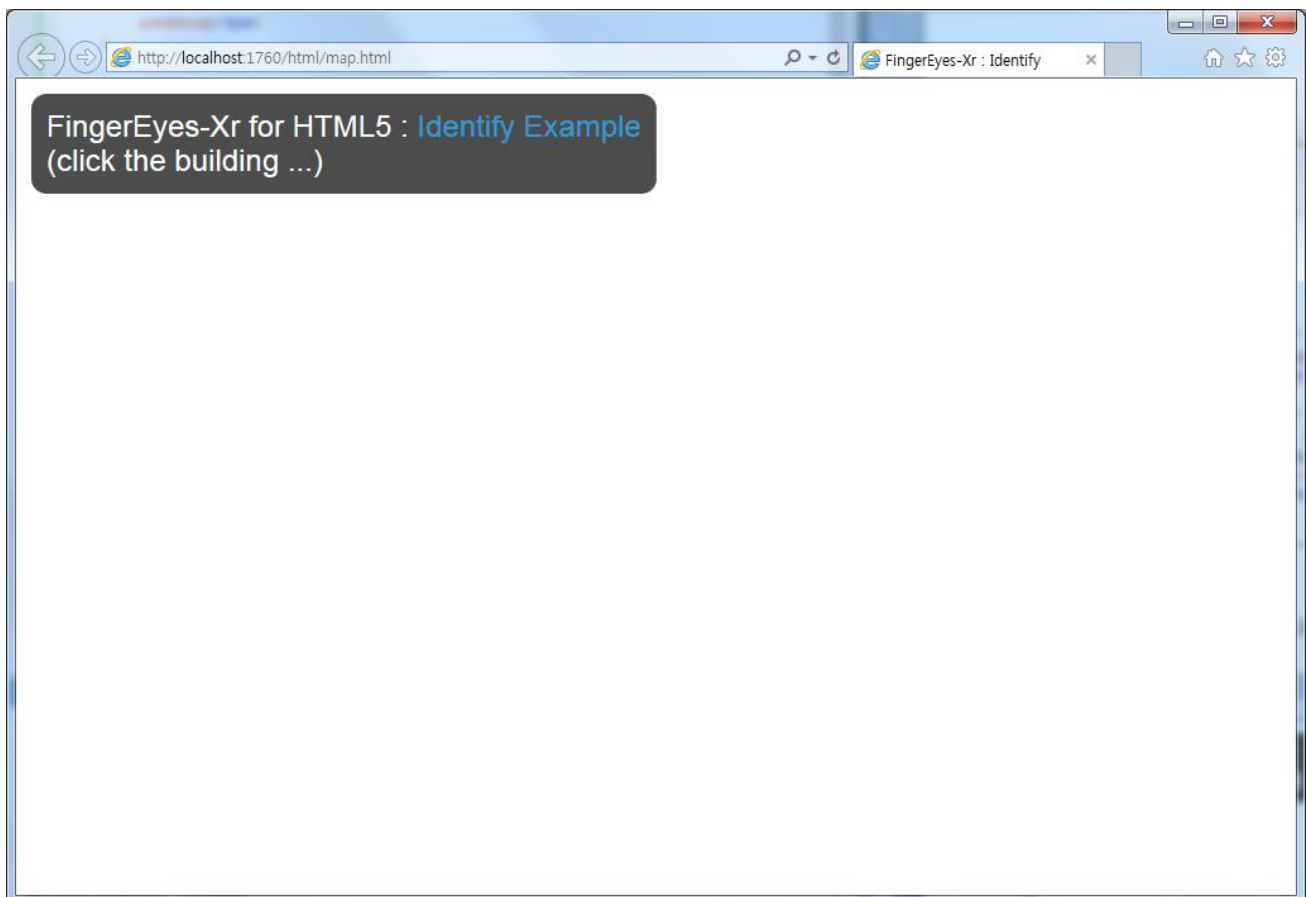


그림 3. 스타일이 적용된 UI

이제 UI와 스타일 적용이 완료되었으므로, 코드를 작성해 보겠습니다. 코드 2. 기본 Script에서 4번 줄에 다음 코드를 추가합니다.

```
01 var map = null;
```

#### 코드 5. DIV와 연결될 지도 객체 정의

이 map 변수는 지도 객체에 대한 참조로 사용됩니다. 이제 <body>에 대한 onload 이벤트를 load() 함수로 지정합니다.

```
01 <body onload="load()" ">
```

#### 코드 6. Body 요소의 onload 이벤트 지정

그리고 load 함수를 다음처럼 추가합니다.

```
01 function load() {
02     map = new Xr.Map("mapDiv", {});
03
04
05
06 }
```

#### 코드 7. Body의 onload 이벤트 함수

2번 코드는 id가 mapDiv인 DIV를 이용하여 지도 객체를 생성하고 이를 앞서 정의해 둔 map 객체에 저장하고 있습니다.

이제 Identify 기능의 대상이 되는 수치지도를 표시하기에 앞서 항공영상 지도를 배경지도로 깔고 그 위에 수치지도를 표시하는 것으로 하겠습니다. 항공영상 지도를 표시는 다음 코드를 통해 가능하며 load 함수의 마지막 부분에 추가합니다.

```
01 var lyr = new Xr.layers.TileMapLayer("basemap",
02     {
03         proxy: "http://222.237.78.208:8080/Xr",
04         url: "http://222.237.78.208:8080/muan_tilemaps",
05         ext: "jpg"
06     }
07 );
```

#### 코드 8. 항공영상 배경지도 레이어 생성

위의 코드는 공간서버로부터 항공영상 지도를 받아와 처리하는 레이어를 생성하는 코드입니다.

이제 수치지도 레이어를 표시 하겠습니다. 수치지도 레이어는 건물을 사용하겠습니다. 사용자가 마우스로 건물을 클릭하면 해당 건물의 속성을 표시하고자 합니다. 다음 코드를 load 함수의 마지막에 추가합니다.

```
01 var bldLyr = new Xr.layers.ShapeMapLayer("bld",
02     {
03         url: "http://222.237.78.208:8080/Xr?layerName=MUAN_BLD"
04     }
05 );
```

#### 코드 9. 수치지도 레이어 생성

위의 코드를 살펴보면, 먼저 1번 코드는 Custom Binary 포맷을 이용하는 수치지도를 사용하기 위해 ShapeMapLayer 객체를 생성하여 bldLyr이라는 변수에 담고 있습니다. 이 클래스의 생성자는 2개의 인자를 갖습니다. 첫 번째는 레이어에 대한 고유한 이름입니다. 이 이름 값을 이용하여 언제든지 해당 이름에 대한 레이어 객체를 참조할 수 있습니다. 두 번째 인자는 Object 타입의 객체로 이 객체에는 url이라는 프로퍼티를 가지고 있어야 합니다. url 프로퍼티를 통해 수치지도 레이어에 대한 연결 문자열(Connection String)을 지정합니다.

이제 앞서 생성한 항공영상 레이어 객체에 대한 lyr과 수치지도 레이어 객체에 대한 bldLyr을 레이어 관리자에 추가해야 합니다. 이에 대한 코드는 아래와 같습니다. load 함수의 마지막 줄에 추가합니다.

```
01 var lm = map.layers();
02
03 lm.add(lyr);
04 lm.add(bldLyr);
```

**코드 10. 생성한 레이어 추가**

이렇게 레이어를 최종적으로 추가 했으니 실행하면 지도가 표시될 것 같지만, 아직 한가지가 더 남아 있습니다. 그것은 레이어를 추가하고 난 뒤에 사용자에게 지도를 표시할 때 표시할 지도의 위치와 축척에 대한 것 입니다. 이에 대한 코드는 다음과 같으며 load 함수의 마지막 줄에 추가합니다.

```
01 map.onLayersAllReady(function () {
02     var cm = map.coordMapper();
03
04     cm.moveTo(151531, 246679);
05     cm.zoomByMapScale(766);
06
07     map.update();
08 });
```

**코드 11. 레이어 추가 완료시 호출될 이벤트 함수 등록**

위의 코드에서 1번의 onLayersAllReady는 앞서 추가한 두 개의 레이어가 문제없이 추가 되면 호출되는 이벤트 함수(Event Function 또는 Callback Function)를 지정하는 함수입니다. 편의상 익명 함수를 사용하여 이벤트를 지정하였습니다. 해당 익명 함수의 코드 부분을 살펴보면, 먼저 2번 코드는 지도의 이동, 확대, 축소, 회전 기능과 지도 좌표와 화면 좌표계 간의 변환 기능을 제공하는 CoordMapper 객체를 가져와 cm 이라는 변수에 저장하고 있습니다. 이 cm 변수를 이용하여 4번 코드에서 화면 상에 표시될 지도의 중심 좌표를 지정하고 5번 코드에서 지도의 축척을 1:766으로 지정하기 위해 축척의 분모값인 766을 인자로 호출하고 있습니다. 최종적으로 7번 코드에서 update 함수를 호출하여 지도를 그리도록 하고 있습니다.



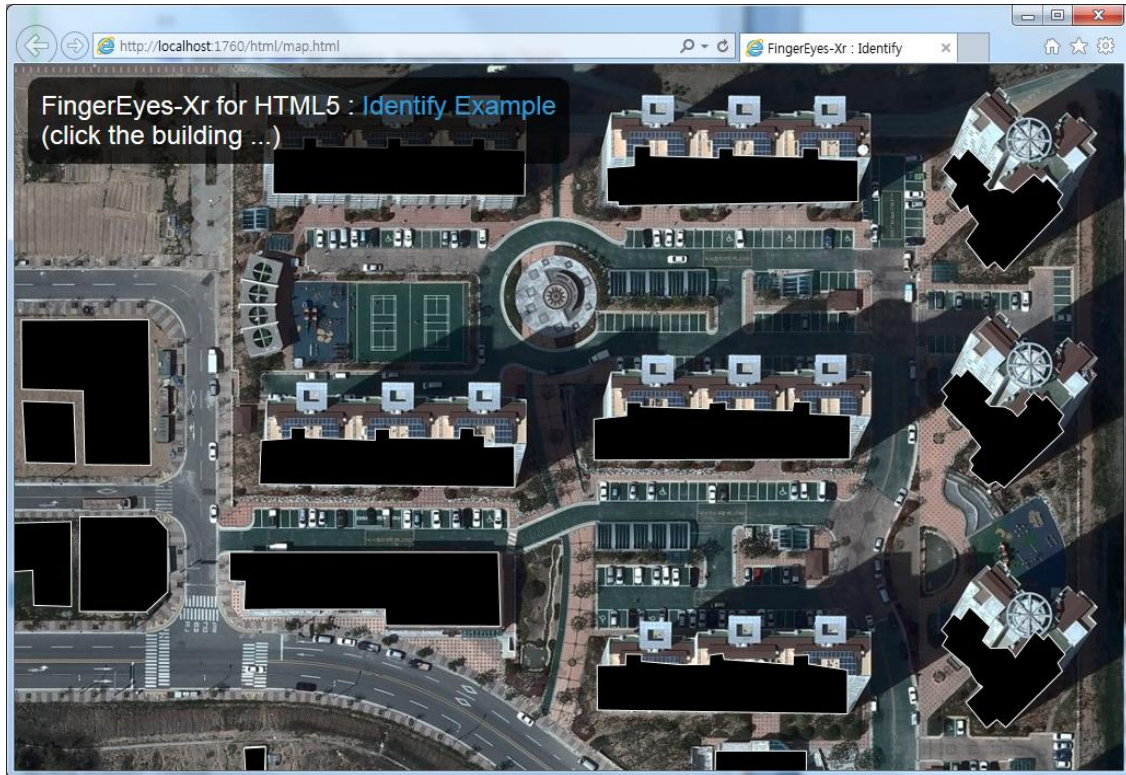


그림 4. 항공영상과 수치지도 레이어 추가에 대한 실행 화면

실행해 보면 건물 도형이 검정색 채움과 하얀색 선으로 표시 되고 있습니다. 건물 도형에 대한 심벌을 변경해 보도록 하겠습니다.

먼저 아래의 코드를 코드 9. 수치지도 레이어 생성 부분의 수치지도 레이어 객체를 생성하는 코드 바로 밑에 추가합니다.

```
01 bldLyr.visibility().visibleByScale(true);
02 bldLyr.visibility().fromScale(0);
03 bldLyr.visibility().toScale(2501);
04
05 var theme = bldLyr.theme();
06 var pen = theme.penSymbol();
07 var brush = theme.brushSymbol();
08
09 pen.color('#ff0000');
10 pen.width(2);
11
12 brush.color('#ffffff');
13 brush.opacity(0.5);
```

코드 12. 수치지도 레이어 그리기 심벌 설정

실행해 보면 다음처럼 건물의 색상 심벌이 변경된 것을 볼 수 있습니다.

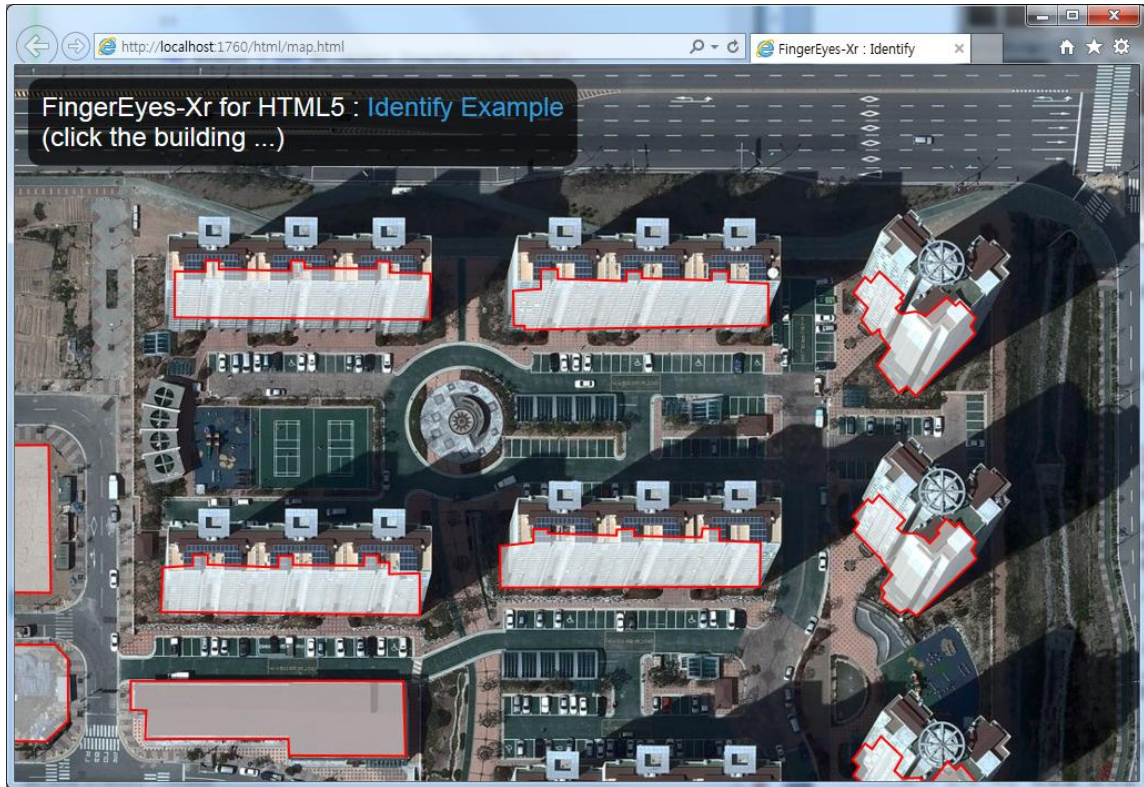


그림 5. 건물의 그리기 심벌 변경

이제 Identify 기능을 위해 아래의 코드를 한줄을 코드 12. 수치지도 레이어 그리기 심벌 설정에서 가장 마지막 줄에 추가해야 합니다.

```
01 bldLyr.needAttribute(true);
```

코드 13. 속성 정보를 항상 서비스 받을 수 있도록 지정하는 코드

이 코드는 해당 수치지도 레이어에 대해서 속성 데이터를 항상 서버로부터 서비스 받으라는 의미입니다. 수치지도는 라벨 표시 등과 같이 속성 데이터를 필요로 하지 않을 경우 속성 데이터를 서버로부터 가져오지 않습니다. 건물에 대한 수치지도 레이어는 속성 데이터를 필요로 하는 라벨 등과 같은 기능을 지정하지 않았으므로 위와 같이 명시적으로 속성 데이터를 가져오라는 코드가 필요한 것입니다.

이제 건물 도형을 클릭할 때 클릭된 도형의 속성을 표시하는 코드를 작성해 보겠습니다. 건물 도형을 클릭했을 때 판별해야 하므로 클릭 이벤트를 등록해야 합니다. 모든 레이어가 완전하게 추가되었을 때 호출되는 이벤트를 등록하기 위해 onLayerAllReady 함수를 사용했습니다. 코드 11. 레이어 추가 완료시 호출될 이벤트 함수 등록을 다시 살펴보면, 이 함수의 인자로 콜백(Callback) 함수를 지정하였는데, 이 콜백 함수의 코드 마지막에 다음과 같은 코드를 추가합니다. 추가된 코드 부분을 *이탤릭체*로 나타내면 아래와 같습니다.

```
01 map.onLayersAllReady(function () {
02     var cm = map.coordMapper();
03
04     cm.moveTo(151531, 246679);
05     cm.zoomByMapScale(766);
```

```

06
07     map.update();
08
09     map.addEventListener(Xr.Events.MapClick, function (e) {
10         var fids = bldLyr.IdByMousePoint(e.viewX, e.viewY, true);
11         if (fids.length > 0) {
12             var id = fids[0];
13             var attribute = bldLyr.attributeById(id);
14             var fieldSet = bldLyr.fieldSet();
15             var cntFields = fieldSet.count();
16             var s = "";
17
18             bldLyr.hilighting(id);
19
20             for (var i = 0; i < cntFields; i++) {
21                 var fName = fieldSet.fieldName(i);
22                 s += "<font face='나눔명조OTF ExtraBold' color='black' size='3'>" +
23                     fName + " " +
24                     "<font face='나눔명조OTF ExtraBold' color='gray' size='3'>" +
25                     attribute.valueAsString(i) + "<br />";
26
27                 if (i == 10) break;
28             }
29
30             var infoWin = new Xr.ui.InfoWindowControl("iwc", map,
31                 new Xr.PointD(e.mapX, e.mapY), s);
32
33             map.userControls().remove("iwc");
34             map.userControls().add(infoWin);
35         } else {
36             //alert("Nonselection");
37         }
38     });
39 });
40

```

코드 14. 클릭된 건물 도형에 대한 속성 정보 표시

새롭게 추가된 코드는 9번째 줄에서부터 39번째 줄까지입니다. 9번은 지도를 클릭했을 때 발생하는 이벤트 리스너(Listener)를 추가하는 코드입니다. 리스너를 추가하는 함수인 `addEventListener`의 첫번째 인자는 등록하고자 하는 이벤트 이름인데, 지도의 클릭 이벤트에 해당하는 `Xr.Events.MapClick` 값이 오고 두번째 인자는 지도를 클릭했을 때 호출되는 콜백함수입니다. 10번 코드에서부터 39번 코드까지가 바로 이 콜백 함수의 실행부입니다. 이 콜백 함수의 실행부를 자세히 설명하도록 하겠습니다.

10번 코드는 사용자가 건물 레이어에 대해서 마우스로 클릭한 지점에 위치하는 건물의 FID 값을 배열로 얻는 함수인 `IdByMousePoint`를 호출하고 있습니다. 이 함수의 첫번째와 두번째 인자는 현재 지도뷰에서 마우스 좌표 X, Y이고 세번째 인자는 클릭한 지점에 위치하는 해당하는 건물 도형을 하나만 가져올지를 지정하는 것으로 `true`를 지정함으로써 해당 위치에 2개 이상의 건물 도형이 있다고 하더라도 첫번째 건물에 해당하는 1개의 건물의 FID 값만을 얻어 오라는 것입니다. 참고로 FID는 수치지도 레이어를 구성하는 도형의 고유 ID 값으로 Feature ID의 축약입니다. 11번 코드는 FID 값이 담긴 배열의 `length`가 0 이상 일 경우, 즉 마우스를 통해 선택된 도형이 있는지를 검사하는 코드입니다. 만약 선택된 도형이 있다면, 12번 코드에서 선택된 도형의 실제 FID 값을 가져오고 가져온 FID 값을 통해 선택된 형의 속성값을 12번 코드를 호출해 얻습니다. 14번과 15번 코드는 속성에 대한 스키마(Schema)에 대한 필드 정보와 필드의 개수를 구하고 있습니다. 15번 코드의 문자열 변수 `s`는 선택된 도형에 대한 속성 정보를 html 형식으로



구성해 담기 위해 사용합니다. 18번 코드는 선택된 도형을 하일라이트(highlighting)시켜 사용자에게 선택된 도형이 어떤 것인지를 시각적으로 부각시켜 줍니다. 20번 코드의 for 문은 선택된 도형의 속성값을 html로 구성해서 변수 s에 저장하고 있습니다. 이 for 문을 보면 28번째 코드에서 i가 10일 때 반복문을 중지시키고 있는데, 필드의 개수가 너무 많을 경우 표시될 내용이 너무 많아지므로 이를 방지하기 위해 11개의 필드 값만을 이용하고 있습니다. 이제 구성된 필드 정보를 정보창(InfoWindow)를 이용해 화면상에 표시하게 되는데, 31번 코드가 정보창을 생성하는 코드로써 첫번째 인자는 정보창의 이름으로써 고유한 ID 값으로 이 ID 값을 이용해 정보창을 코드를 이용해 닫을 수 있습니다. 그리고 두번째 인자는 정보창이 호출될 지도 객체입니다. 그리고 세번째 인자는 정보창이 지도 객체에 표시될 좌표값입니다. 그리고 네번째가 정보창에 표시될 html 형식의 문자열 값입니다. 이 문자열 값은 앞서 구성한 문자열 변수 s에 해당합니다. 34번 코드는 혹시 이전에 정보창이 표시되고 있다면 표시된 정보창을 닫으라는 것이고 35번은 새롭게 정보창을 표시하라는 코드입니다.

이제 실행하고 건물 중 하나를 클릭하면 다음 화면처럼 선택된 건물의 속성 정보가 표시되는 것을 볼 수 있습니다.

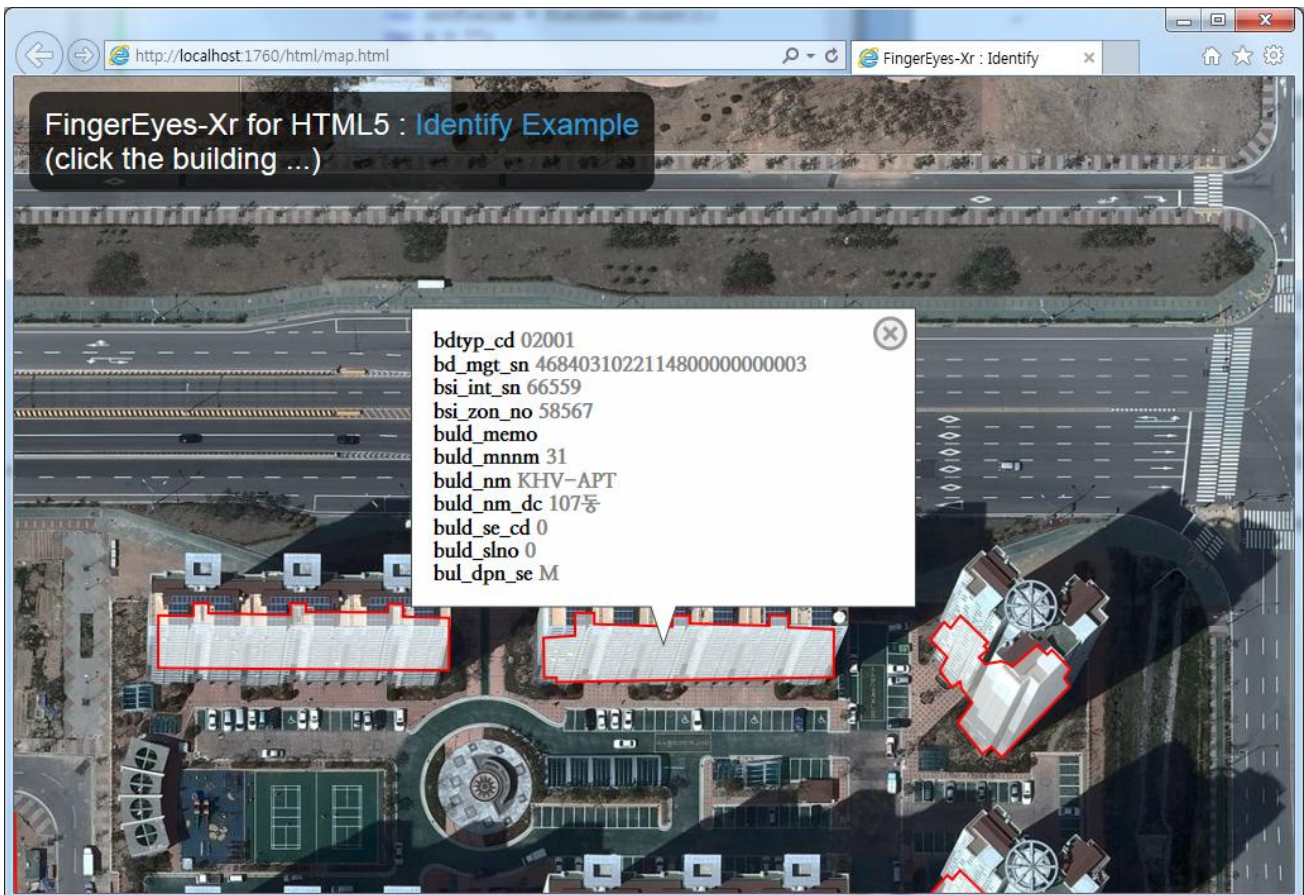


그림 6. 선택된 건물 도형의 속성 정보 표시

이상으로 수치지도 레이어를 구성하는 도형을 클릭하고, 클릭된 도형의 속성 정보를 지도 상에서 정보창을 통해 표시하는 Identify 기능에 대한 설명을 마칩니다.

지금까지 작성한 코드 전체는 아래와 같습니다.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <style>
    body
    {
      margin:0px;
      padding:0px;
    }

    #mainLayout
    {
      position:relative;
      width:100%;
      height:100%;
      border:none;
    }

    #mapDiv {
      top:0px;
      left:0px;
      position:absolute;
      width:100%;
      height:100%;
      border:none;
      overflow:auto;
    }

    #title {
      top:12px;
      left:12px;
      padding: 12px;
      position:absolute;
      background:rgba(0,0,0,0.7);
      border:none;
      overflow:auto;
      border-radius: 12px;
      font-size: 24px;
      color: #ffffff;
      font-family: "Arial";
    }
  </style>

  <title>FingerEyes-Xr : Identify</title>

  <script src="http://www.geoservice.co.kr/z/Xr.min.1.0.js"></script>
  <script type="text/javascript">
    var map = null;

    function load() {
      map = new Xr.Map("mapDiv", {});

      var lyr = new Xr.layers.TileMapLayer("basemap",
      {
        proxy: "http://222.237.78.208:8080/Xr",
        url: "http://222.237.78.208:8080/muan_tilemaps",
        ext: "jpg"
      }
      );

      var bldLyr = new Xr.layers.ShapeMapLayer("bld",
      {
```

```

        url: "http://222.237.78.208:8080/Xr?layerName=MUAN_BLD"
    }
);

bldLyr.visibility().visibleByScale(true);
bldLyr.visibility().fromScale(0);
bldLyr.visibility().toScale(2501);

var theme = bldLyr.theme();
var pen = theme.penSymbol();
var brush = theme.brushSymbol();

pen.color('#ff0000');
pen.width(2);

brush.color('#ffffff');
brush.opacity(0.5);

bldLyr.needAttribute(true);

var lm = map.layers();

lm.add(lyr);
lm.add(bldLyr);

map.onLayersAllReady(function () {
    var cm = map.coordMapper();

    cm.moveTo(151531, 246679);
    cm.zoomByMapScale(766);

    map.update();

    map.addEventListener(Xr.Events.MapClick, function (e) {
        var fids = bldLyr.IdByMousePoint(e.viewX, e.viewY, true);
        if (fids.length > 0) {
            var id = fids[0];
            var attribute = bldLyr.attributeById(id);
            var fieldSet = bldLyr.fieldSet();
            var cntFields = fieldSet.count();
            var s = "";

            bldLyr.hilighting(id);

            for (var i = 0; i < cntFields; i++) {
                var fName = fieldSet.fieldName(i);
                s +=
                    "<font face='나눔명조OTF ExtraBold' color='black' size='3'>" +
                    fName + " " +
                    "<font face='나눔명조OTF ExtraBold' color='gray' size='3'>" +
                    attribute.valueAsString(i) + "<br />";

                if (i == 10) break;
            }

            var infoWin = new Xr.ui.InfoWindowControl(
                "iwc", map, new Xr.PointD(e.mapX, e.mapY), s);

            map.userControls().remove("iwc");
            map.userControls().add(infoWin);
        } else {
            //alert("Nonselection");
        }
    });
});

```

```
        });  
    });  
  
    }  
  
    </script>  
</head>  
<body onload="load()">  
    <div id="mainLayout">  
        <div id="mapDiv"></div>  
        <div id="title">  
            FingerEyes-Xr for HTML5 :  
            <font color="#349bd6">Identify Example</font>  
            <br />  
            (click the building ...)  
        </div>  
    </div>  
</body>  
</html>
```