

웹 접근성 적용 가이드

1. 개요

본 가이드는 웹 접근성 평가지표를 준수하는 웹 응용프로그램을 개발하기 위해 eXbuilder6에서 지원하는 기능들을 설명합니다. 웹 접근성 평가지표는 2017년 정보접근성 지침이 및 멘토 양성교육의 교재를 근거로 하고 있으며 가이드 내용은 지속적으로 보완 및 수정될 수 있습니다.

2. 지표 별 적용가이드

2.1. 적절한 대체 텍스트 제공

(적절한 대체 텍스트 제공) 텍스트 아닌 콘텐츠는 그 의미나 용도를 이해할 수 있도록 대체 텍스트를 제공하고 있다.

준수기준 : 의미 있는 텍스트가 아닌 콘텐츠에 그 의미나 용도를 동등하게 인식할 수 있는 적절한 대체 텍스트를 제공해야 한다.

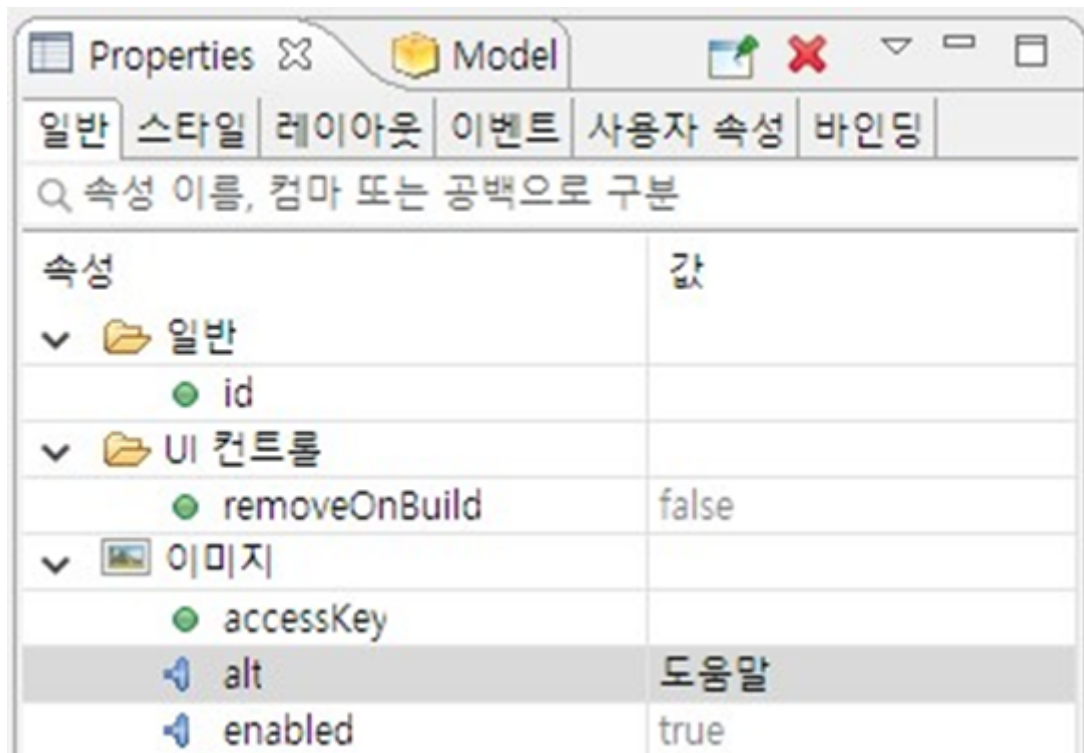
주의사항 : 컨트롤 속성의 tooltip 으로 표현되는 title 속성은 대체텍스트로 인정되지 않으며 title 속성의 값이 alt 속성과 동일한 내용일 경우 스크린리더에 의해 내용이 중복으로 인식되어 사용자에게 불편을 주게 되므로 alt 와 tooltip 속성은 구분해서 사용해야 한다.

2.1.1. 텍스트 아닌 콘텐츠에 대한 대체 텍스트 제공

1) 시각적으로 인지가 가능한 이미지 텍스트에 이미지와 동등한 정보를 alt 속성을 통해 제공한다

- 이미지 컨트롤의 alt 속성을 사용하여 대체 텍스트를 제공한다

<그림 1. 이미지 컨트롤의 alt 속성>



- alt 속성 사용 시 다음과 같은 결과가 나타난다

<소스 1. alt 속성 사용 결과>

```

```

2) 이미지맵의 img, area 요소에 alt 미정의 오류인 경우

[3.6.1.1 이미지맵 처리] 예제 참조

3) longdesc 파일이 없는 경우

- longdesc 속성은 미지원

4) 의미 있는 이미지가 배경이미지로 제공되었으나 대체 콘텐츠가 제공되지 않은 경우

- 정보를 담고 있는 의미 있는 이미지에 해당 이미지의 정보를 alt 속성을 통해 설정할 수 있다.

❗ 배경 이미지에는 alt 속성이 사용 불가합니다. eXbuilder6에 맞춰 작성이 필요합니다.

[3.1.1.1 대체 텍스트 처리] 예제 참조

<그림 2. 의미 있는 이미지>



- 의미 있는 이미지에 대체 텍스트를 제공한 결과를 확인할 수 있다

<소스 2. 기호에 대한 대체 콘텐츠>

```



```

5) 특수문자 또는 기호 등으로 상승/하락, 상한/하한, 더보기 등의 정보를 제공하거나 ASCII 심벌 기호를 이용하여 얼굴 형상을 표현하거나, 한글 자음 또는 모음을 나열하여 감정을 표현하는 문자 이모티콘을 심벌 또는 문자로만 나열하여 제공한 경우

- alt 속성을 이용하여 순위 정보 또는 주가의 상승 및 하락에 대한 내용을 나타낼 수 있다 Alt 속성은 요소에만 사용 가능합니다. eXbuilder6에 맞춰 변경해야 합니다. 하단 소스는 의 대체 텍스트 제공 방법을 나타냅니다.

[3.1.1.1 대체 텍스트 처리] 예제 참조

<소스 3. alt에서의 특수기호 사용>

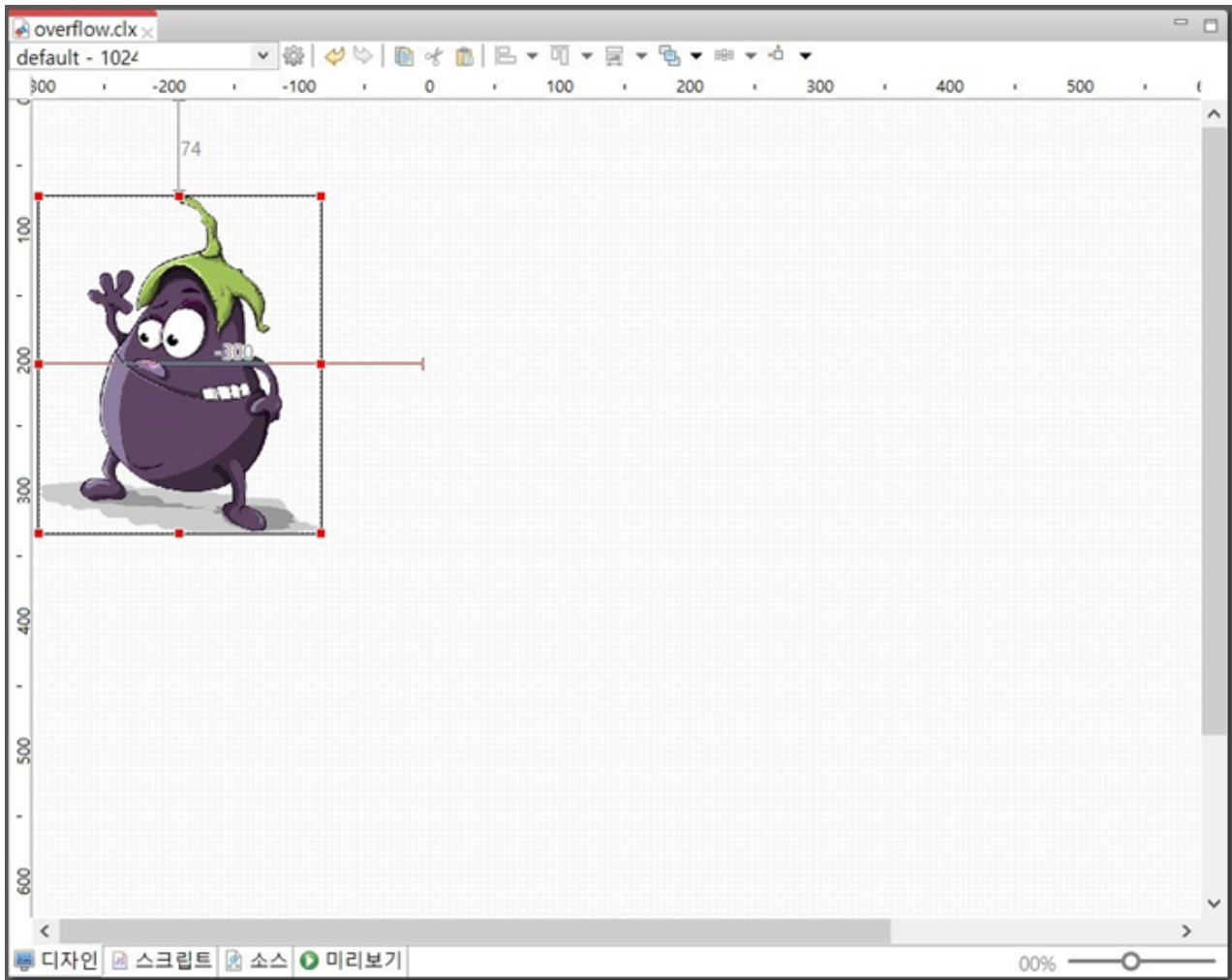
```

```

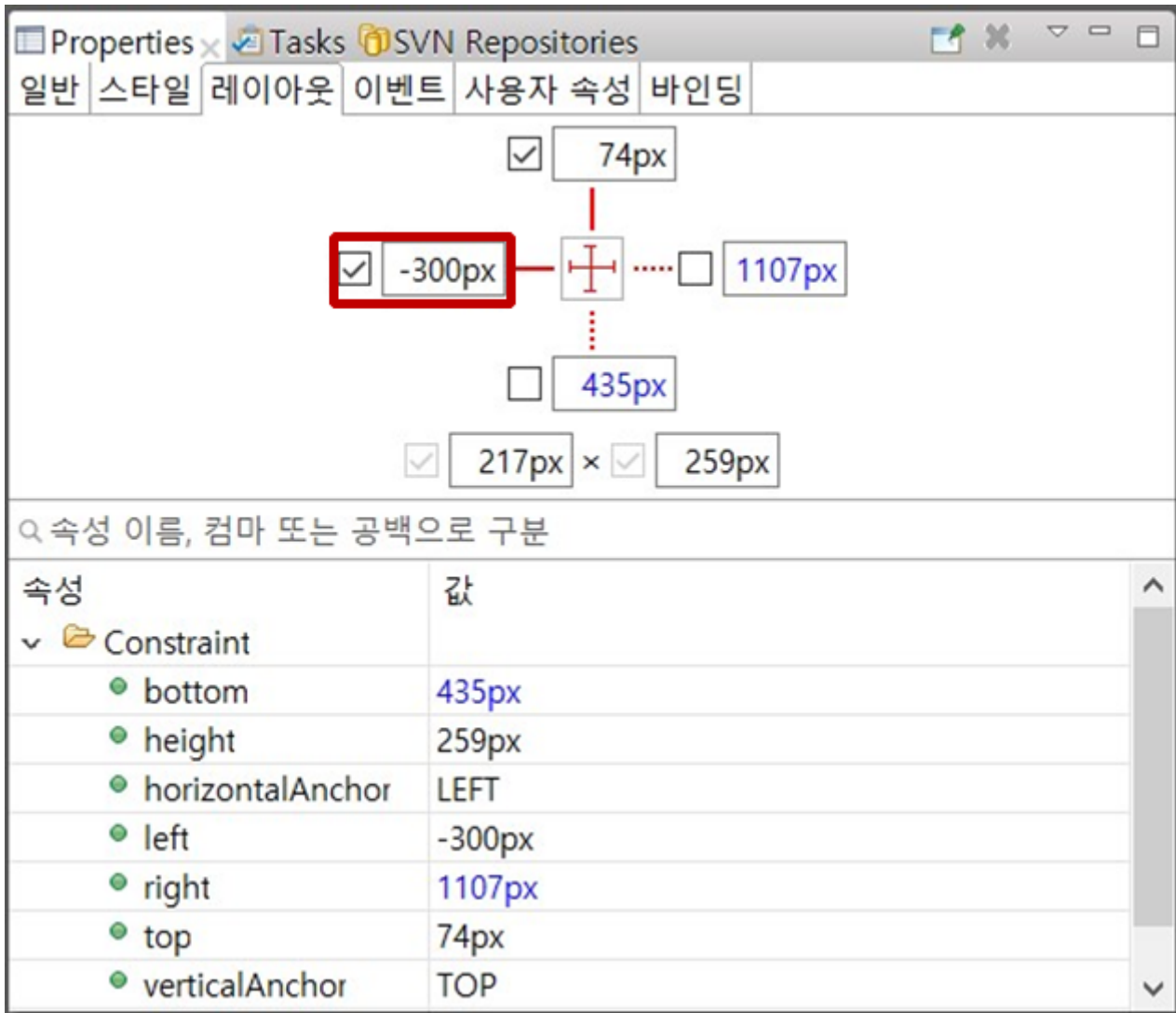
6) 이미지 요소에 IR 기법 제공시 display:none 또는 visibility:hidden 속성을 사용한 경우

- display: none 이나 visibility:hidden 을 사용하지 않고 포지셔닝 기법으로 접근성 있게 대체 텍스트를 제공할 수 있다
- overflow : hidden 과 이미지 요소의 위치를 변경하여 해결이 가능하다

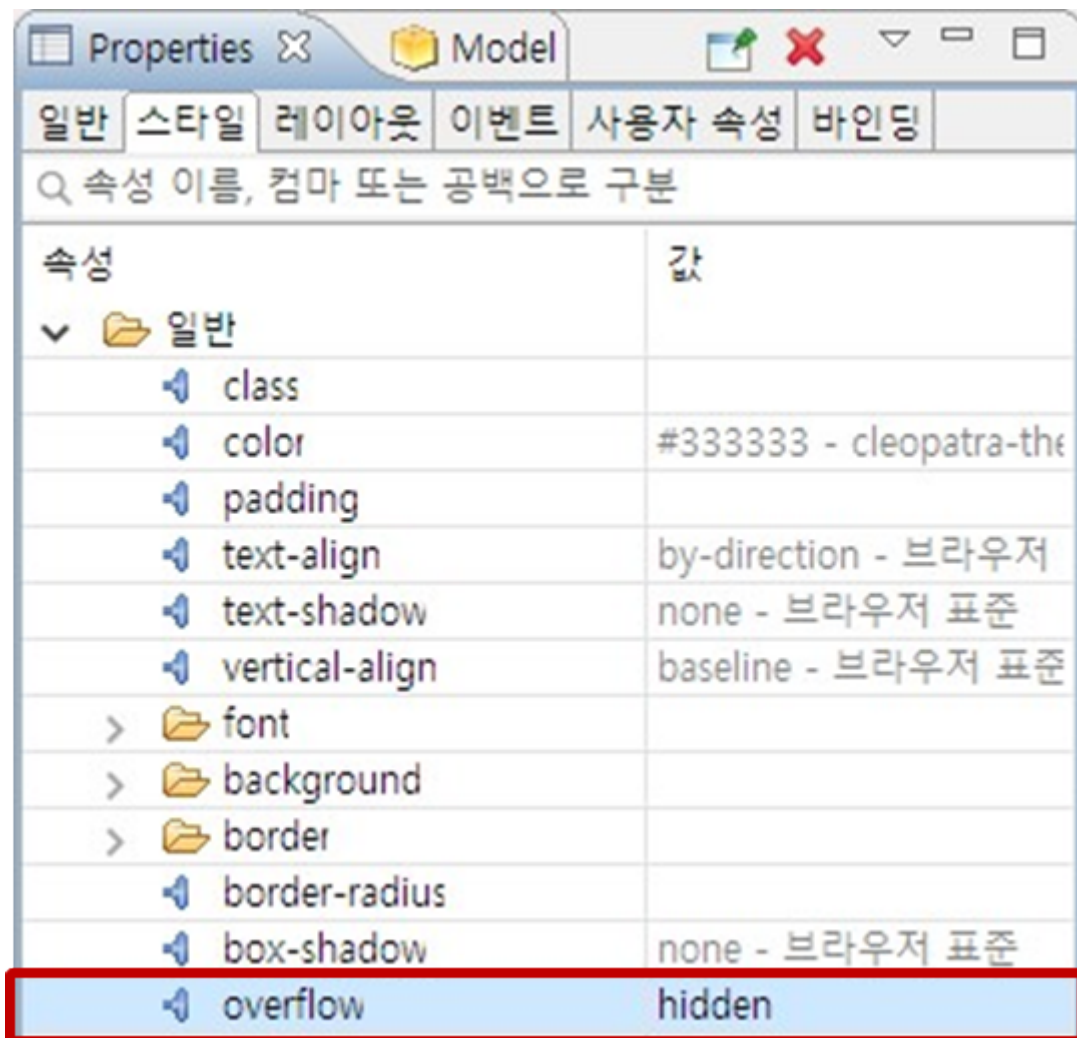
<그림 3. IR 기법 제공을 위한 이미지 컨트롤 배치>



<그림 4. 이미지 컨트롤 앵커>



<그림 5. 포지셔닝 기법 사용>



- 앵커와 overflow 속성을 지정하여 화면에서 벗어나 사용자에게 보이지 않지만 다음과 같은 결과를 확인할 수 있다

<소스 4. overflow:hidden 지정 결과>

```
<div class="cl-control cl-image"
  style="overflow:hidden;position:absolute;top:74px;left:300px;width:217px;height:259px"
  data-ndid="93d46457-de49-df13-e497-3b1314e8fa50" role="img" tabindex="-1"
  id="uuid-2a17266b-b77b-3155-c0e1-8a8a57461e36" ariadisabled="false">

</div>
```

2.1.2. 텍스트 아닌 콘텐츠에 대한 대체 텍스트를 적절하게 제공

- 대체텍스트를 제공하는 aria-label 속성이 무분별하게 사용되면 스크린리더 사용시 콘텐츠 탐색에 오류를 야기할 수 있다.

- 의미없이 제공되는 aria-label="" 과 title="" 같이 빈 값으로 제공되는 속성은 스크린리더에 혼란을 줄 수 있기 때문에 eXbuilder6는 빈 값으로 설정된 tooltip이나 fieldLabel 속성은 aria-label이나 title속성으로 생성하지 않는다.

1) 대체텍스트의 정보가 불충분하게 제공된 경우, 대체텍스트가 오타 또는 업데이트가 되지 않은 경우, 조직도, 도식 이미지, 차트 등 복잡한 이미지의 대체 텍스트는 단순 텍스트 제공 이상의 상호 관계, 수식, 흐름 등 충분한 설명을 제공해야 함

[3.1.1.1 대체 텍스트 처리] 예제 참조

2) 장식 또는 꾸밈의 목적으로 제공된 의미 없는 이미지(또는 배경이미지)에 대체텍스트를 제공한 경우, 이미지 맵의 img 요소에 대체텍스트가 빈 값(alt="") 또는 잘못 제공된 경우

- 의미없이 제공되는 alt="" 과 같이 빈 값으로 제공되는 속성은 스크린리더에 혼란을 줄 수 있기 때문에 이미지컨트롤의 alt 속성을 빈 값으로 설정한 경우 이미지컨트롤은 alt 속성을 생성하지 않는다
의미가 없는 이미지는 alt=""로 대체텍스트 속성을 제공해야합니다.

3) longdesc 연결되지 않는 경우

- longdesc 속성은 미지원

4) QR 코드, CCTV 등 실시간 영상 등에 대한 대체 텍스트는 콘텐츠의 용도나 목적을 알 수 있도록 제공

- QR 코드를 이미지 컨트롤을 통해 나타내고 이미지 컨트롤의 alt, tooltip 속성을 통해 대체 텍스트를 제공할 수 있다
Title 속성으로는 대체텍스트가 인정되지 않습니다.

<그림 6. 이미지에 표시된 QR 코드>



- alt 속성에 URL을 입력하여 대체텍스트를 제공할 수 있다

<소스 5. QR 코드의 URL 입력>

```

```

- HTML 스니펫 컨트롤을 사용하여 QR 코드 이미지에 사용한 목적이나 용도를 대체 텍스트로 제공하고 이동할 수 있는 링크도 제공할 수 있다

<소스 6. HTML 스니펫으로 대체텍스트 제공>

```
<a href = "http://m.site.naver.com/0q16M">
  
</a>
```

[3.6.1.2 <a> 태그 처리] 예제 참조

5) 자동 가입 방지 이미지(CAPTCHA)에 대하여 대체 텍스트를 빈 값이나 동등한 텍스트 정보로 제공하지 않은 경우 음성 등의 대체 수단을 제공해야 함

- 이미지 컨트롤의 src를 CAPTCHA를 제공하는 서버 URL로 설정하면 이미지 컨트롤 내에서 CAPTCHA이미지를 표현할 수 있다.

- 추가로 음성듣기 버튼을 제공하여 대체 수단을 제공할 수 있다.

<그림 7. 자동 가입 방지 이미지 대체텍스트 제공>



2.1.3. 사용자가 등록한 텍스트가 아닌 콘텐츠에 대해 관리자가 대체 텍스트 등록을 유도하는 안내 또는 대체 수단 입력 기능을 제공

- 사용자가 업로드한 이미지 파일을 화면에 출력하는 경우 대체 텍스트 값을 업로드하는 파일과 함께 입력 받아 화면에 출력할 때 alt에 값을 바인딩하여 표현할 수 있다

[3.1.1.2 바인딩] 예제 참조

2.1.4. 기타

1) alt 와 title 값을 동일하게 제공한 경우

- alt 속성에는 대체텍스트를 작성하고, tooltip 속성에는 부가 설명을 작성할 수 있다
(alt 와 tooltip 이 동일한 내용인 경우 리더에서 같은 내용이 두 번 읽히기 때문에 유의해야 한다)

[3.1.1.1 대체 텍스트 처리] 예제 참조

2) 썸네일로 사용되는 단순 이미지가 해당 텍스트와 함께 링크에 포함될 경우 alt 중복설명 제공된 경우, 링크를 포함하며 텍스트 없이 단독으로 제공되는 이미지에 은 제공되고 있으나 이미지의 alt를 빈 값으로 제공한 경우

[3.6.1.2 <a> 태그 처리] 예제 참조

3) 플래쉬로 제작된 콘텐츠의 대체텍스트

- 해당 없음

2.2. 자막 제공

(자막 제공) 멀티미디어 콘텐츠에는 자막, 원고 또는 수화를 제공해야 한다.

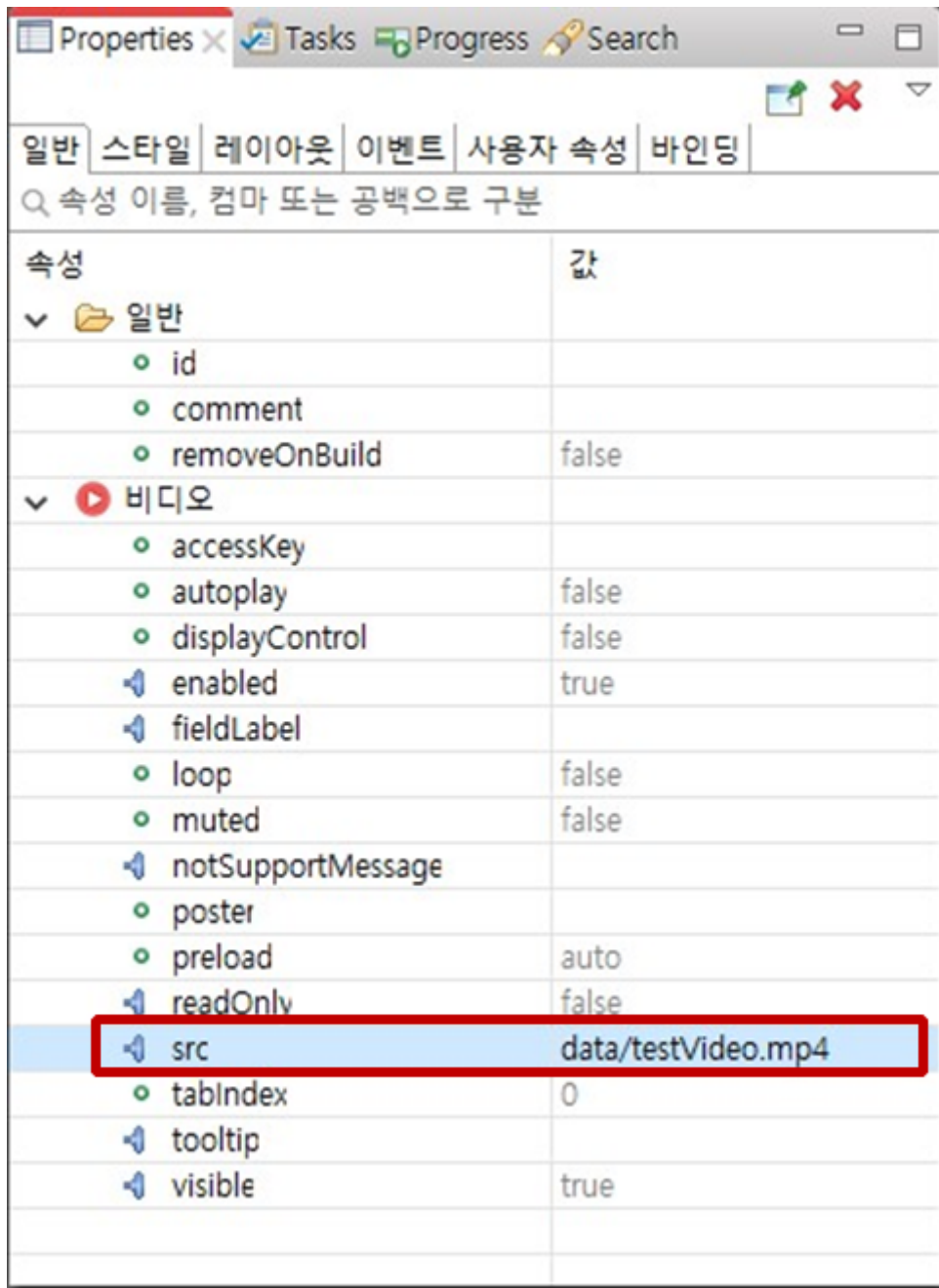
준수기준 : 영상, 음성 등 멀티미디어 콘텐츠를 동등하게 인식할 수 있는 자막, 원고 또는 수화를 제공해야 한다

2.2.1. 멀티미디어 콘텐츠에 자막, 원고, 수화 중 하나 이상의 대체 수단을 제공

1) 멀티미디어 콘텐츠에 대한 대체수단을 제공하지 않은 경우

- 디자인 편집기에 비디오 컨트롤을 이용하여 멀티미디어 콘텐츠를 배치할 수 있다

<그림 8. src 속성 지정>



위의 속성을 지정하면 다음과 같은 결과가 나타난다.

<소스 7. 비디오의 src 속성 설정>

```
<video class="cl-media" style="width:100%; height:100%" data-ndid="bec07beb-af23-c164-74ba-7064db111783" src="clx/2019/R-34/data/testVideo.mp4" tabIndex="0"></video>
```

- 읽을 수 있는 자막은 아웃풋컨트롤과 혼합된 UDC를 개발하여 표시할 수 있다.

- 비디오 컨트롤과 혼합하여 사용하는 아웃풋 컨트롤에 ariaLive 속성을 사용하여 자막을 실시간으로 인식할 수 있다.
- 별도의 버튼(자막 표시/숨김)을 사용하여 자막 표시상태를 수정할 수 있게 한다.
- 자막이 표시되지 않을 경우 아웃풋 컨트롤의 ariaLive속성을 none(default)로 설정한다.(aria-live : 실시간으로 내용을 갱신하여 스크린리더에서 음성 피드백으로 알립니다. 값으로 polite, assertive, off(default)를 설정할 수 있으며 갱신하는 내용의 중요도에 따라 선택합니다.)

<그림 9. 리더로 읽을 수 있는 자막 컨트롤>



2) 음성이 없는 동영상(시각적 정보만 제공)에 대체수단이 없는 경우(ppt, 슬라이드쇼, 동영상 등)

- 비디오컨트롤 또는 HTML 오브젝트를 이용한 UDC를 개발하여 제공할 수 있다

3) 배경음만 재생되는 단순 동영상에 대체수단이 없는 경우

- 영상의 설명을 요약하여 아웃풋 컨트롤 또는 HTML 스니펫 컨트롤로 제공할 수 있다

4) 파일로 제공되는 영상, 음성 콘텐츠에 대체수단이 제공되지 않은 경우

[3.6.1.2 <a> 태그 처리] 예제 참조

2.2.2. 멀티미디어 콘텐츠에 대한 대체 수단의 내용을 적절하게 제공

1) 멀티미디어 콘텐츠의 내용을 일부 누락하거나 요약하여 제공하여서는 안됨, 멀티미디어 콘텐츠에 대한 대체 수단 제공 방식이 적절하지 않은 경우 오류

- 영상의 설명을 요약하여 아웃풋 컨트롤 또는 HTML 스니펫 컨트롤로 제공할 수 있다

2.2.3. 사용자가 등록한 멀티미디어 콘텐츠에 대해 관리자가 대체 수단 등록을 유도하는 안내 또는 대체 수단 입력 기능을 제공

1) 멀티미디어 콘텐츠에 대한 대체 수단 제공 방식이 적절하지 않은 경우 오류

- 해당 없음

2.3. 색에 무관한 콘텐츠 인식

(색에 무관한 콘텐츠 인식) 콘텐츠는 색에 관계없이 인식될 수 있어야 한다.

준수기준 : 색을 배제하여도 인식할 수 있는 콘텐츠를 제공해야 한다

- 웹 컨트롤을 통해 차트를 제공할 때 차트의 표현은 색상에만 의존하지 않고 무늬 등을 통해 구분할 수 있도록 제공되어야 한다

2.4. 명확한 지시 사항 제공

(명확한 지시 사항 제공) 지시사항은 모양, 크기, 위치, 방향, 색, 소리 등에 관계없이 인식될 수 있어야 한다.

준수기준 : 지시 사항 정보를 특정 감각에 의존하지 않고 다양한 감각을 통해 용도나 목적을 이해할 수 있도록 제공해야 한다

2.4.1. 색, 크기, 모양, 방향 등 지시사항을 시각정보 외에 별도 안내 내용과 함께 제공

- 아웃풋 컨트롤을 배치하여 value 속성을 통해 지시사항에 대해 상세하게 작성할 수 있다

[3.3.1 입력 계열 컨트롤의 레이블 활용 가이드] 참조

2.4.2. 소리 등 지시사항을 청각정보 외에 별도 안내내용과 함께 제공

[2.4.1 색, 크기, 모양, 방향 등 지시사항을 시각정보 외에 별도 안내 내용과 함께 제공] 참조

2.5. 텍스트 콘텐츠의 명도 대비

(텍스트 콘텐츠의 명도 대비) 텍스트 콘텐츠와 배경 간의 명도 대비는 4.5 대 1 이상이어야 한다.

준수기준 : 내비게이션(메뉴) 및 본문 콘텐츠에서 텍스트의 명도 대비가 4:5:1 이상 이거나, 18pt 또는 굵은 14pt 이상에서 3:1 이상이어야 한다

2.5.1. 일반 텍스트 또는 이미지 텍스트의 명도 대비는 3:1

- 콘텐츠의 명도는 Properties의 ‘스타일’ 탭에서 색상을 수정하여 대비를 만족할 수 있도록 할 수 있다

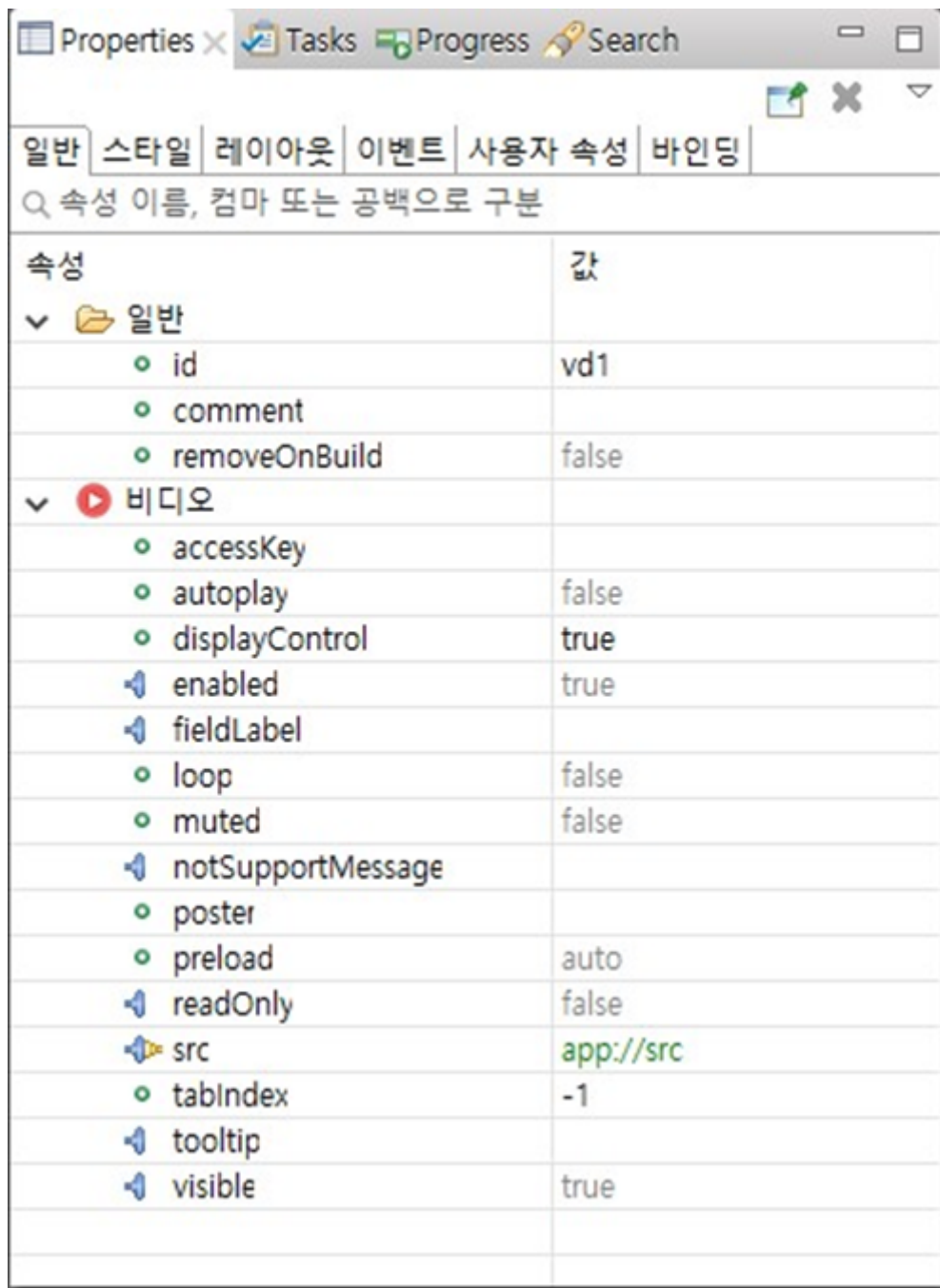
2.6. 자동 재생 금지

(자동 재생 금지) 자동으로 소리가 재생되지 않아야 한다.

준수기준 : 배경음 사용하지 않았거나 3 초 이내 또는 바로 정지 할 수 있는 배경음 콘텐츠를 사용해야 한다

- 자동으로 재생되는 경우, 비디오 컨트롤의 displayControl 속성을 true 로 설정하면 재생을 멈추거나 소리를 끌 수 있는 기능을 표시할 수 있다

<그림 10. 비디오 컨트롤의 조작 기능 표시>



2.7. 콘텐츠 간의 구분

(콘텐츠 간의 구분) 이웃한 콘텐츠는 구분될 수 있어야 한다.

준수기준 : 이웃한 콘텐츠가 테두리, 구분선, 서로 다른 무늬 등을 이용하여 시각적으로 구분되도록 제공되어야 한다

2.7.1. 이웃한 콘텐츠를 테두리를 이용하여 구분

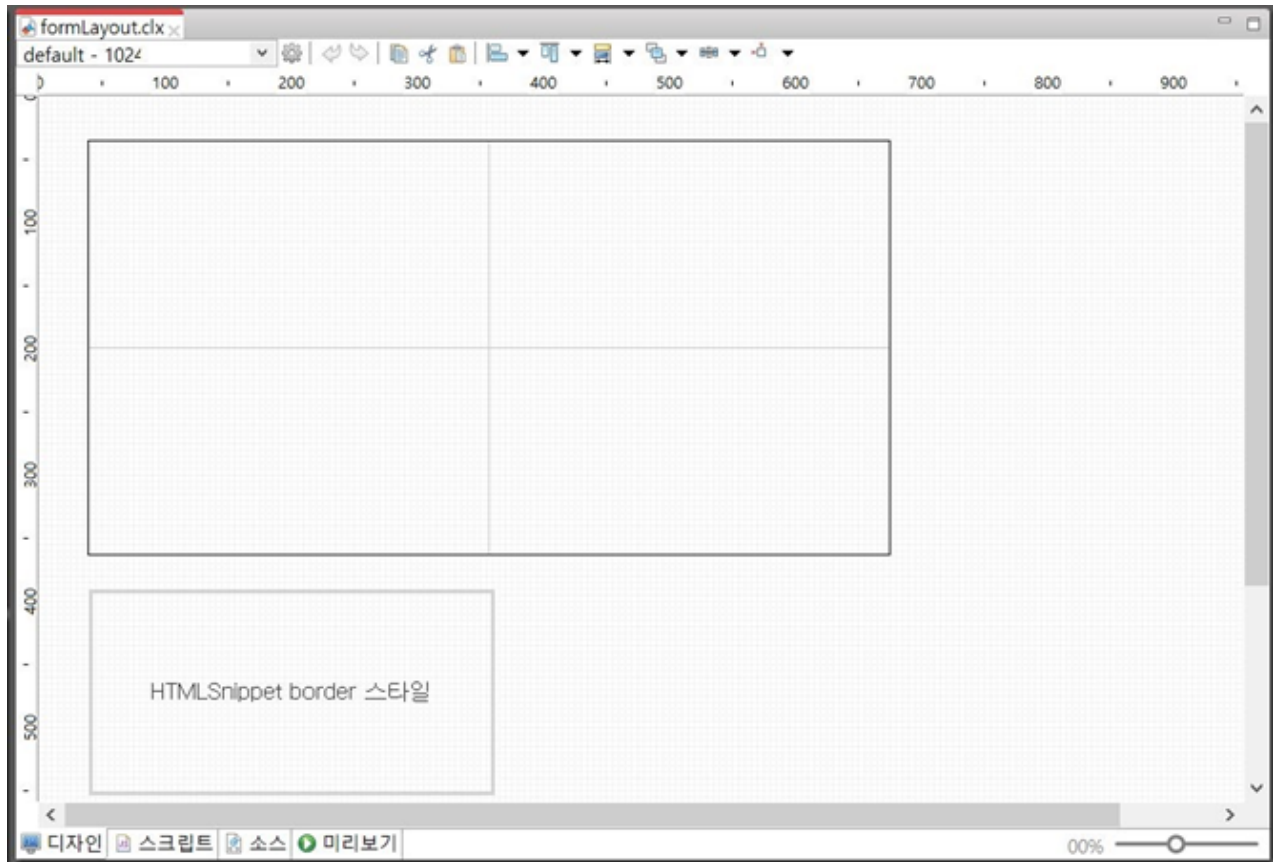
- 이웃한 콘텐츠를 아래 1 가지 방법 이상으로 구분하지 않은 경우 테두리를 이용하여 구분

1. 컨트롤의 보더 스타일을 이용하여 구분함
2. 콘텐츠 사이에 시각적인 구분선을 삽입하여 구분함
3. 서로 다른 무늬를 이용하여 구분함
4. 콘텐츠 배경색 간의 명도대비(채도)를 달리하여 구분함
5. 줄 간격 및 글자 간격을 조절하여 구분함
6. 기타 콘텐츠를 시각적으로 구분할 수 있는 방법 등을 이용하여 구분함

1) 이웃한 콘텐츠는 서로 같은 수준의 콘텐츠로 저 시력 사용자도 구분할 수 있을 정도로 구분해주는 것을 권장, 테두리, 구분 선, 콘텐츠 배경 색 간의 명도 대비, 기타 시각적으로 구분할 때 구분을 위한 색과 배경색의 명도 대비가 4.5:1 이상이 되도록 하는 것을 권장

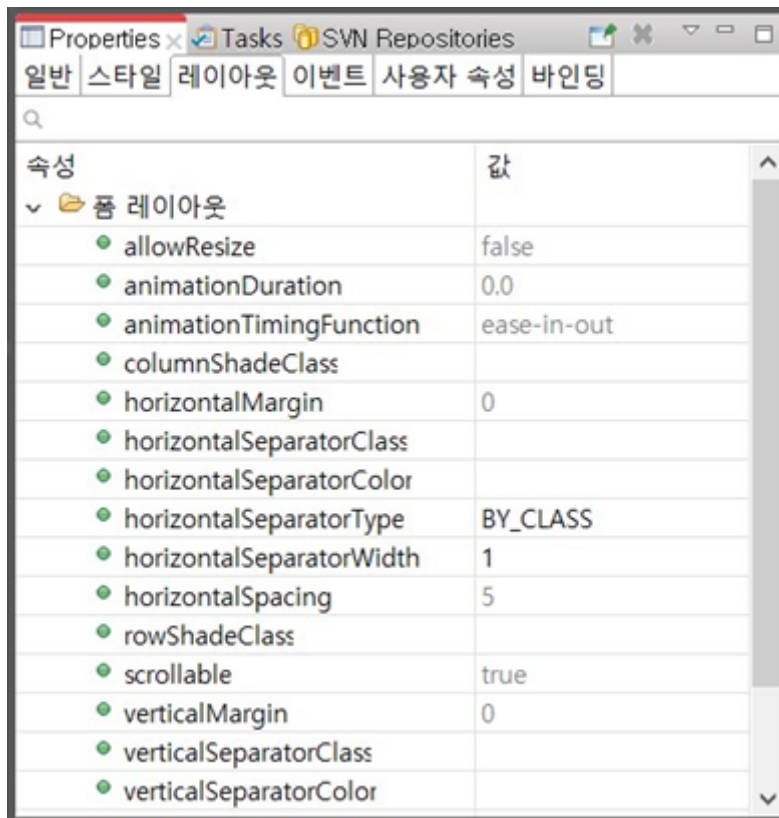
- 각 컨트롤의 Properties의 ‘스타일’ 탭에서 ‘border’, ‘background-color’ 등의 속성을 수정하여 구분할 수 있다
- 폼 레이아웃의 구분선을 설정할 경우, Properties의 ‘레이아웃’ 탭에서 설정할 수 있다

<그림 11. 폼레이아웃과 HTML 스니펫 컨트롤의 border 스타일>



- 폼레이아웃을 선택한 뒤 Properties의 '레이아웃' 탭에서 구분선의 스타일을 설정할 수 있다
- 컨트롤의 border는 '스타일' 탭의 border 속성에서 설정할 수 있다

<그림 12. 폼 레이아웃 구분선 스타일 설정>



2.8. 키보드 사용 보장

(키보드 사용 보장) 모든 기능은 키보드만으로도 사용할 수 있어야 한다.

준수기준 : 모든 기능이 키보드로 접근 및 운용 가능하도록 제공되어야 한다

2.8.1. 마우스로 제어할 수 있는 요소를 키보드로 제어 가능

1) 마우스 이벤트만 적용된 경우

- eXbuilder6의 컨트롤은 마우스 이벤트와 대응되는 키보드 이벤트를 제공한다

<표 1. 컨트롤의 마우스, 키보드, 포커스 이벤트>

컨트롤	이벤트			비고
	마우스	키보드	포커스	
HTML 스니펫	○	○	○	HTML 스니펫은 포커스를 가지지 않으나 내부 태그 요소는 포커스를 가질 수 있음. 단 tabIndex 속성을 통한 임의 포커스 순서 제어는 지원하지 않고 태그의 순서대로만 포커스가 이동함.

HTML 오브젝트	X	X	X	
MDI 폴더	○	○	○	헤더영역이 포커스를 가 집
탭 폴더	○	○	○	헤더영역이 포커스를 가 집
그룹	○	○	X	그룹은 포커스를 가지 지 못하나 포함한 컨트 롤은 포커스를 가질 수 있음
그리드	○	○	○	
아웃풋	○	X	X	tabIndex를 지정하면 포 커스를 받을 수 있음
인풋박스	○	○	○	buttonFocusable 속성 값이 true(기본값 false) 인 경우 삭제 버튼도 포 커스를 받을 수 있음
넘버에디터	○	○	○	
데이트인풋	○	○	○	
마스크에디터	○	○	○	
텍스트에리어	○	○	○	
서치인풋	○	○	○	buttonFocusable 속성 값이 true(기본값 false) 인 경우 검색, 삭제 버튼 도 포커스를 받을 수 있 음
리스트박스	○	○	○	
링크드리스트박스	○	○	○	
콤보박스	○	○	○	
링크드콤보박스	○	○	○	
체크박스	○	○	○	
체크박스그룹	○	○	○	
임베디드앱	○	○	X	임베디드앱은 포커스를 가지지 못하나 포함한 컨 트롤은 포커스를 가질 수 있음
임베디드페이지	X	X	X	
다이얼로그	○	○	○	다이얼로그 타이틀과 닫 기 버튼도 포커스를 받 을 수 있음
이미지	○	○	○	
버튼	○	○	○	
라디오버튼	○	○	○	
메뉴	○	○	○	
트리	○	○	○	
네비게이션바	○	○	○	
캘린더	○	○	○	
셸	X	X	X	
슬라이더	○	○	○	

알림	X	X	X	
아코디언	○	○	○	아코디언의 헤더영역이 포커스를 가짐
파일업로드	○	○	○	
파일인풋	○	○	○	
페이지인덱서	○	○	○	
프로그레스	○	○	X	

- mouseover 이벤트에 자바스크립트를 적용한 경우 대응될 수 있는 키보드 이벤트인 focus를 함께 적용한다

2) 이미지 또는 초점을 받을 수 없는 요소에 onClick 이벤트가 적용된 경우, 이미지 또는 초점을 받을 수 없는 요소에 마우스 이벤트만 적용되어서는 안된다

[3.2.4 onclick 이벤트 및 onkeydown 이벤트] 예제 참조

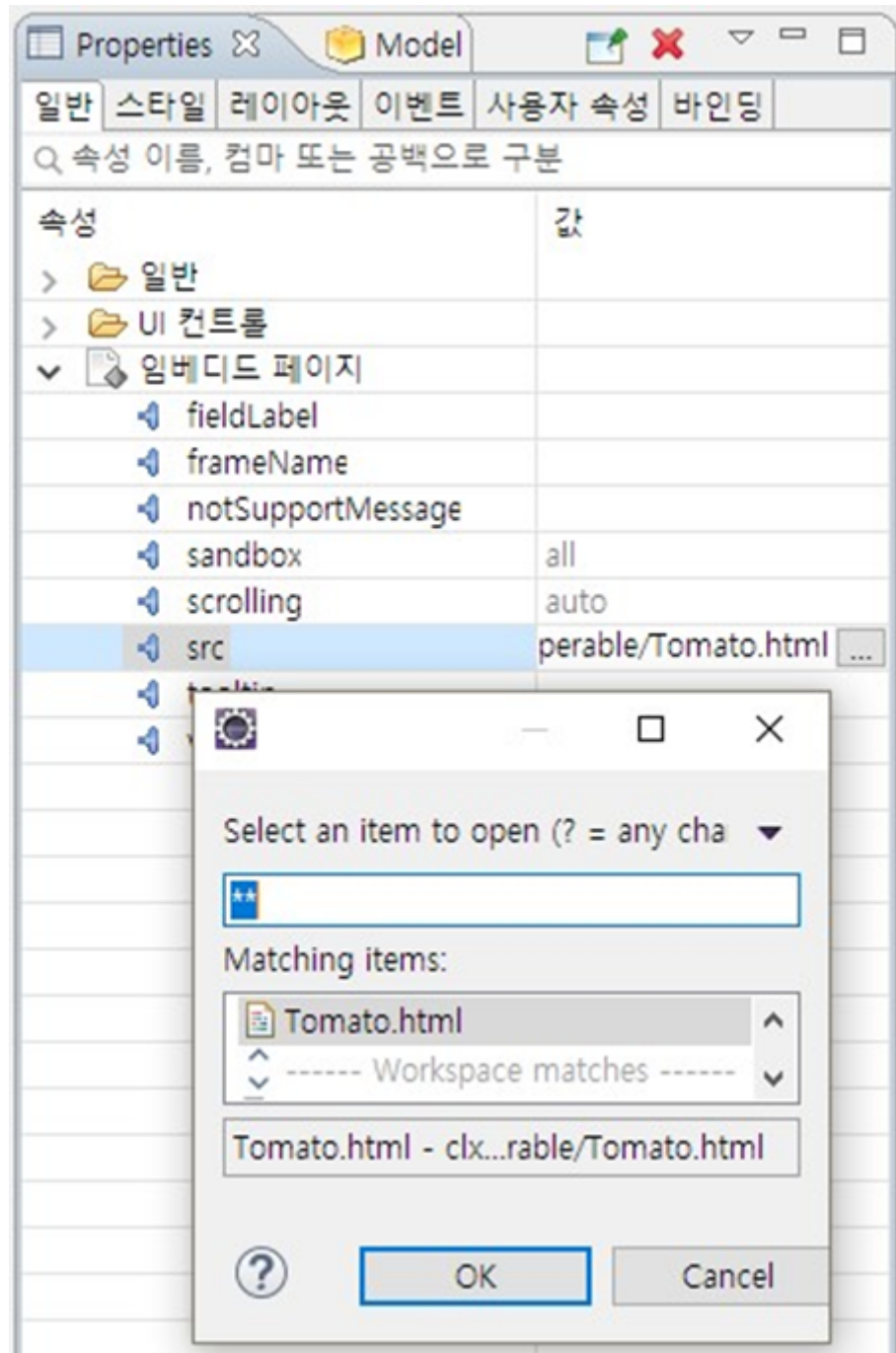
3) a 요소에 href 속성이 제공되지 않아 키보드 접근 및 초점이 표시되지 않는 경우

- href가 포함된 a 요소를 HTML 스니펫 컨트롤 혹은 임베디드 페이지를 통하여 제공할 수 있다
- HTML 스니펫 컨트롤의 value 속성에 a 태그를 작성한 후 tab 키를 통해 초점을 이동할 수 있다

<소스 8. 임베디드 페이지에 표시할 HTML - src>

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF8">
<title>토마토시스템</title>
</head>
<body>
  <a href="http://tomatosystem.co.kr/solution/#">토마토시스템</a>
</body>
</html>
```

<그림 13. 임베디드 페이지로 a 태그 구현>



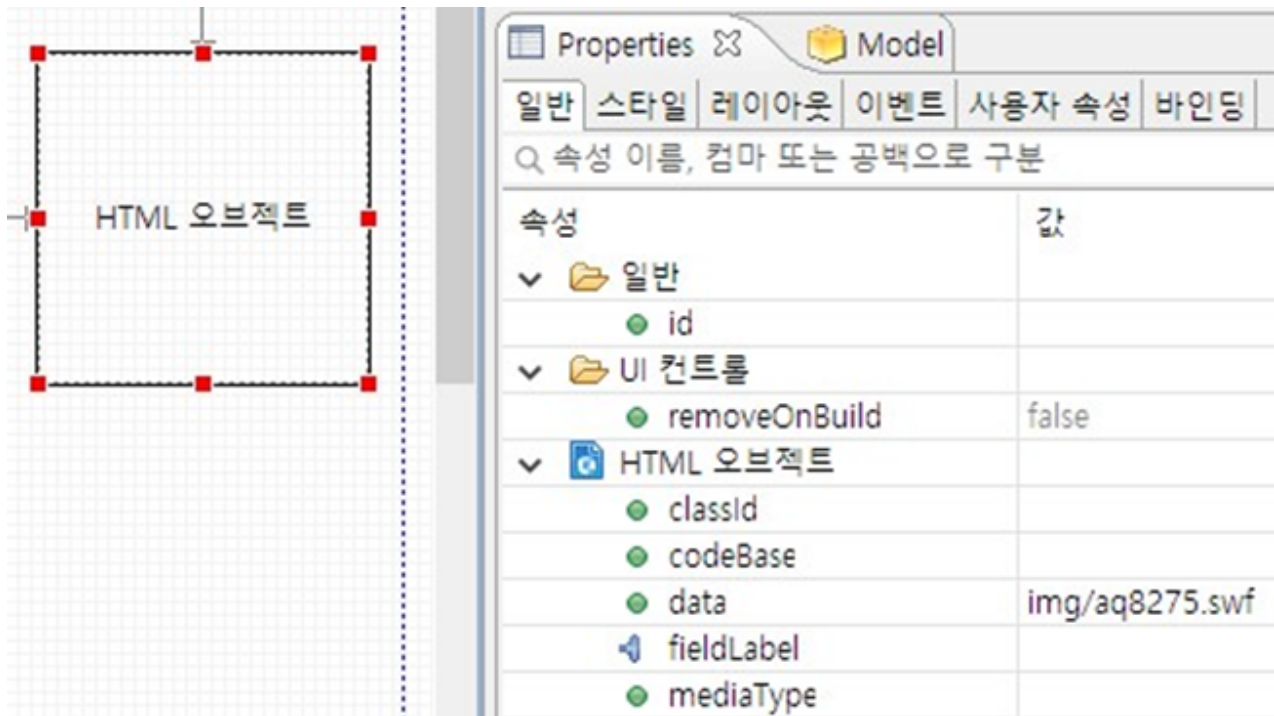
<그림 14. 임베디드 페이지 구현 결과>

토마토시스템

4) 키보드 포커스가 필요한 플래쉬 콘텐츠의 wmode를 transparent, opaque 로 제공하는 경우

- HTML 오브젝트 혹은 임베디드 페이지를 통해 플래쉬를 제공할 수 있다

<그림 15. HTML 오브젝트로 플래쉬 구현>



이때, 플래쉬 콘텐츠의 wmode 속성을 transparent, opaque 로 설정하지 않는다

<소스 9. wmode의 transparent, opaque 오류 예시>

```
<object type="application/x-shockwave-flash" data="../img/aq8275.swf" width="100%" height="auto">
  <param name="movie" value="../img/aq8275.swf" />
  <param name='wmode' value='transparent' />
  <param name='allowScriptAccess' value='always' /> </object>
```

5) 키보드 포커스가 필요 없는 플래쉬 콘텐츠의 wmode를 window 로 제공하는 경우

- 플래쉬의 wmode 속성을 window 로 제공하지 않는다

<소스 10. wmode의 window 오류 예시>

```
<object type="application/x-shockwave-flash" data="../img/aq8275.swf" width="100%" height="auto">
  <param name="movie" value="../img/aq8275.swf" />
```

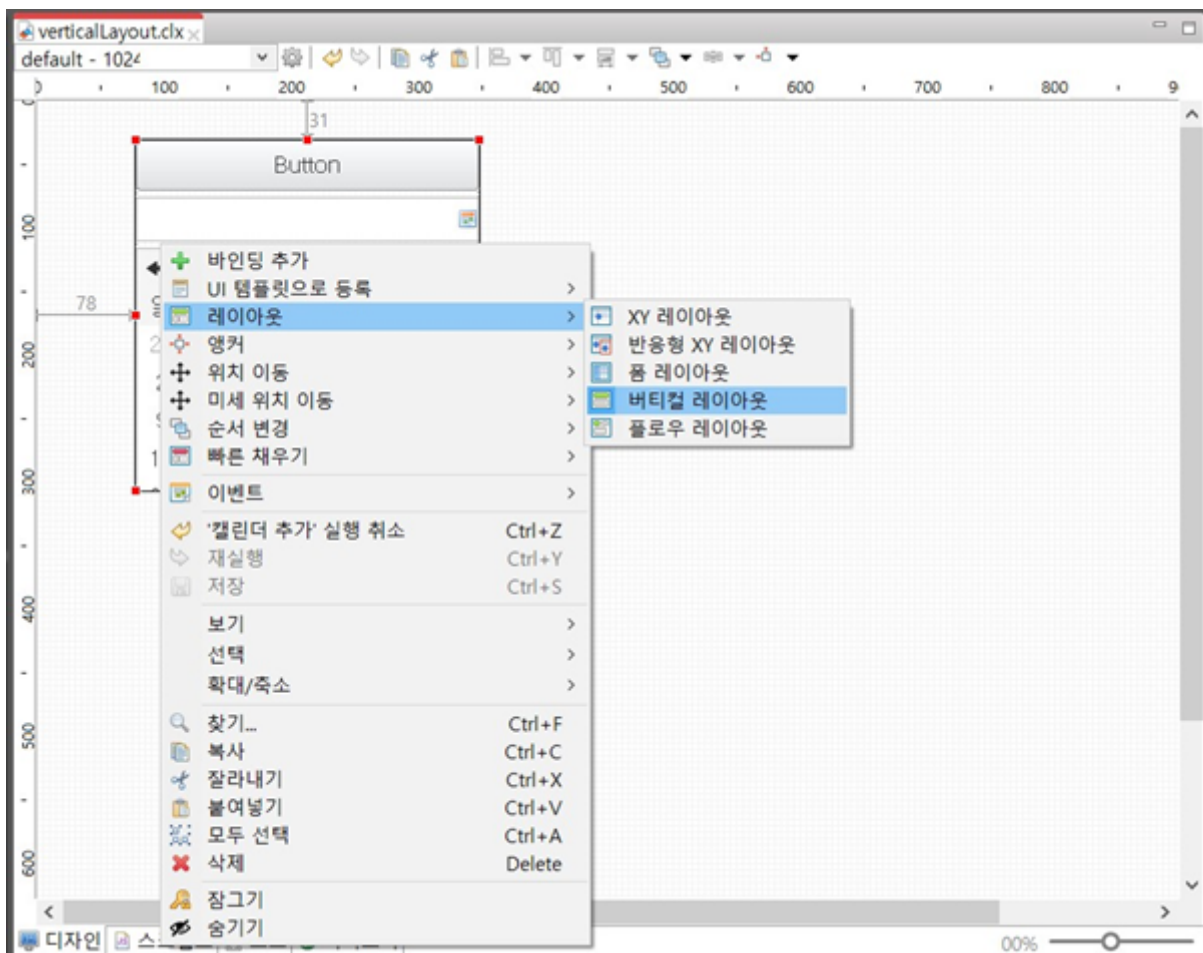
```
<param name='wmode' value='transparent' />
<param name='allowScriptAccess' value='always' /> </object>
```

6) 스크롤 되는 콘텐츠는 키보드로도 이용할 수 있어야 함

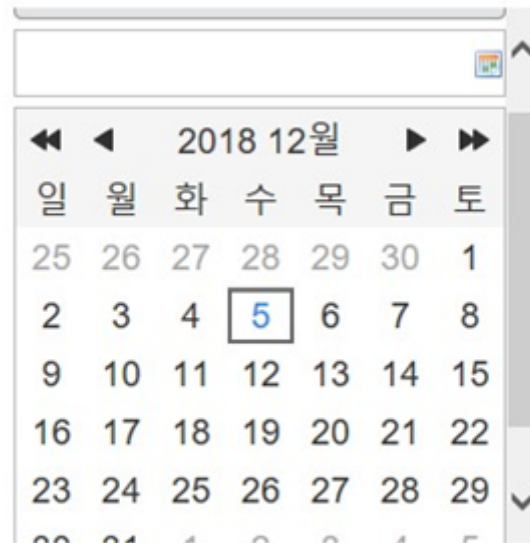
- 스크롤이 있는 컨트롤에 포커스가 있는 상태일 때 방향키를 사용하여 스크롤의 위치를 변경할 수 있다

[3.2.1. 초점 제어] 예제 참조

<그림 16. vertical-layout 설정>



<그림 17. vertical-layout 설정 결과>



<그림 18. 그리드 컨트롤의 스크롤>

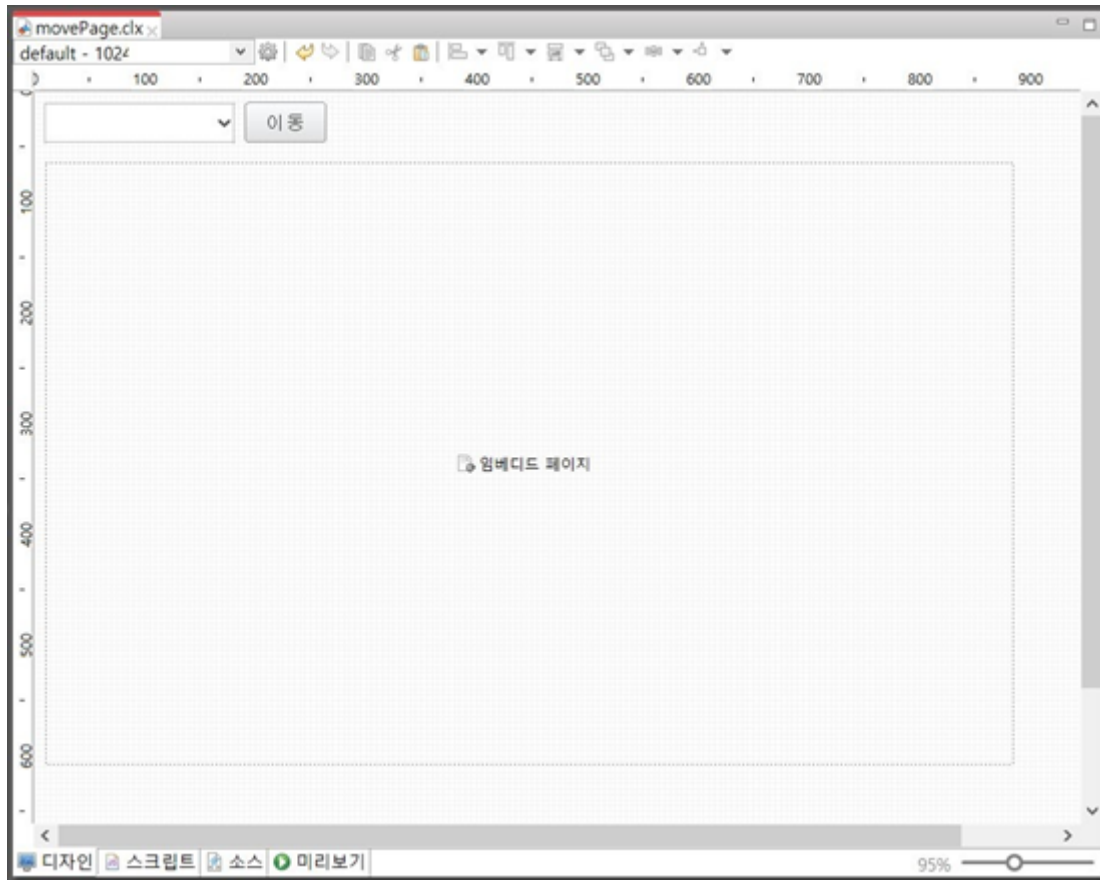
열1	열2	열3
column14	column24	column34
column15	column25	column35
column16	column26	column36
column17	column27	column37
column18	column28	column38
column19	column29	column39

- 그리드 컨트롤은 화살표 키를 사용하여 행 이동을 통해 스크롤을 이동할 수 있다 7) select에 onChange를 사용하여 select option 값을 선택하는 것만으로 페이지가 새롭게 변경되는 경우

[3.2.3 컨트롤의 내부 이동 단축키] 예제 참조

- 콤보박스의 selection-change 이벤트에 페이지가 새롭게 변경되도록 구현하는 것이 아니라, select에서 값 선택 후 버튼 컨트롤을 통해 이동될 수 있도록 구현한다
- 버튼 컨트롤을 클릭하면 하단의 임베디드 페이지 컨트롤을 통해 페이지가 이동할 수 있다

<그림 19. 페이지 이동을 위한 컨트롤 배치>



<소스 11. 버튼 클릭했을 때 페이지 이동 이벤트>

```
/*
 * "이동" 버튼에서 click 이벤트 발생 시 호출.
 * 사용자가 컨트롤을 클릭할 때 발생하는 이벤트.
 */
function onBtn1Click( /* cpr.events.CMouseEvent */ e) {
    /**
     * @type cpr.controls.Button
     */
    var btn1 = e.control;
    var ep = app.lookup("ep1");

    ep.src = "http://www.tomatosystem.co.kr";
}
```

위의 내용을 실행하면 다음과 같은 결과를 확인할 수 있다.

<그림 20. 페이지 이동 결과>



2.8.2. 기타

1) 반복영역 건너뛰기 링크에 키보드 접근이 불가능한 경우

- 디자인 특성상 건너뛰기 링크를 보이지 않도록 구현하는 경우에도 키보드 접근에 의해 초점을 받으면 건너뛰기 링크가 표시되어야 하므로 CSS를 사용하여 구현 가능하다

<소스 12. 건너뛰기 링크 구현 예시>

```
<a href="http://tomatosystem.co.kr/solution/#"
  style="display: block; overflow: hidden; height: 1px; width: 1px; margin:0 -1px -1px 0;
  font-size:0; lineheight:0">토마토시스템</a>
```

[3.4.1 네비게이션바 컨트롤을 이용한 반복영역 건너뛰기] 예제 참조

2) onFocus="this.blur();" 사용 오류

- focus 이벤트에 app.lookup(ControllId).blur() API 를 사용하지 않는다

3) 스크립트 사용으로 인한 키보드 함정에 빠진 경우

- 메뉴를 제공하는 네비게이션 바는 키보드 방향키로 메뉴 간 이동을 제공하고 탭 키는 다른 컨트롤로 초점을 이동시켜 초점 이동이 메뉴에 갇히는 것을 방지한다

[3.2.1 초점 제어] 예제 참조

2.9. 초점 이동

(초점 이동) 키보드에 의한 초점은 논리적으로 이동해야 하며 시각적으로 구별할 수 있어야 한다. 준수기준 : 운용 가능한 개체(링크, 버튼, 서식 등)들에 키보드로 접근 시 초점을 인지할 수 있고 논리적 순서로 이동되어야 한다

2.9.1. 초점의 이동 순서를 논리적이고 일관성 있게 제공

1) 초점의 이동순서(문서구조)가 논리적이지 않으며 일관성이 없는 경우

- 키보드의 접근 순서를 논리적 순서로 제공해야 하기 때문에 tabIndex를 지정할 때 순차적으로 지정할 수 있도록 한다

i tabIndex는 팔레트 뷰의 도구(Tab Index)를 사용하면 편리하게 지정이 가능하다

- 복합적인 구조를 가진 UI에서 tabIndex의 지정은 자칫 의도치 않게 논리적 모순이 발생할 수 있기 때문에 tabIndex 를 통한 초점 이동 보다는 컨트롤의 생성 순서를 조절하여 자동으로 초점이 이동될 수 있도록 지정하는 것을 권장한다

i UDC 로 구현된 컨트롤 내부의 tabIndex 와 UDC를 포함하는 외부 App의 tabIndex간의 모순

[2.8.1 마우스로 제어할 수 있는 요소를 키보드로 제어 가능] 참조

2) 초점의 이동순서(로그인, 검색 등 기능을 담고 있는 콘텐츠)가 논리적이지 않으며 일관성이 없는 경우

[3.2.1 초점 제어] 예제 참조

3) 초점의 이동순서(제목과 콘텐츠가 분리됨)가 논리적이지 않으며 일관성이 없는 경우

- Outline 뷰에서 컨트롤의 생성 순서를 논리적인 초점 이동 순서에 맞게 지정한다

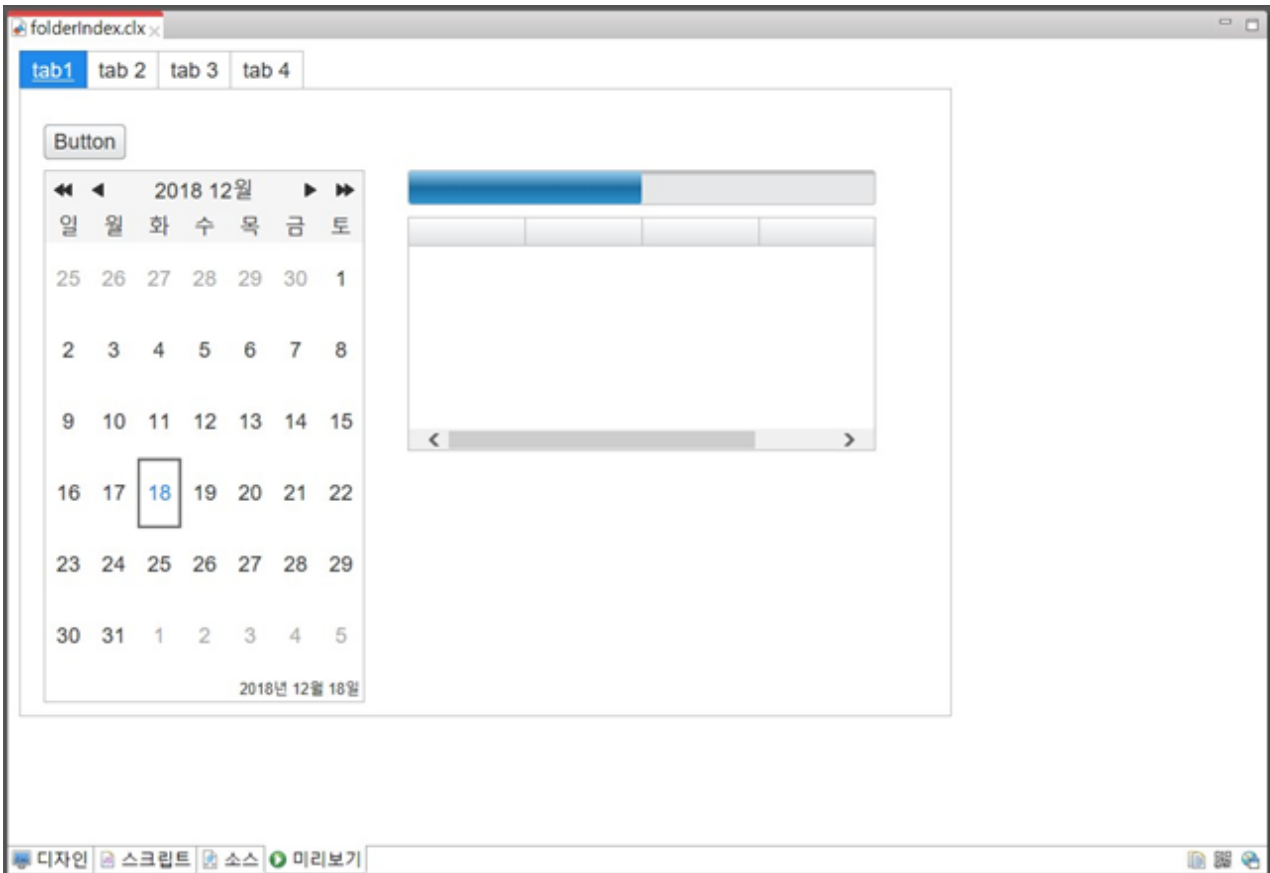
[3.2.1 초점 제어] 예제 참조

4) Tab 콘텐츠에서 제목과 내용이 display:none, block 으로 구분 되었을 때 초점이동이 논리적이지 않은 경우

- TabFolder, MDIFolder 컨트롤의 TabItem 생성시, TabItem의 순서대로 초점이동이 가능하다
 - 탭컨트롤은 좌/우 방향으로 탭버튼 간에 이동되고 선택된 탭아이템에서 탭키를 통해 탭 내부 콘텐츠로 이동한다
- 아코디언은 키보드 탭키를 통해 선형화된 순서로 초점이 이동한다
 - 아코디언 타이틀 A -> 아코디언 A 의 내부 콘텐츠 -> 아코디언 타이틀 B -> 아코디언 B 의 내부 콘텐츠
- 아코디언은 상/하 방향키를 통해 아코디언 타이틀 간의 초점이동이 가능하다

[3.2.3 컨트롤의 내부 이동 단축키] 예제 참조

<그림 21. TabFolder 컨트롤의 TabItem 초점 이동>



5) 페이지 로딩 시 제공되는 화면을 가리는 레이어 팝업 또는 모달창

- app.openDialog()를 통하여 화면 전체를 가리는 다이얼로그(팝업)을 여는 경우 닫기 버튼을 별도로 두어 다이얼로그를 닫을 수 있도록 구현한다
- 다이얼로그를 닫을 수 있도록 1) 다이얼로그 Prop에서 headerClose를 true로 설정한다. 2) 다이얼로그에서 headerClose를 false로 설정하였을 경우에는 다이얼로그를 닫을 수 있는 컨트롤을 추가하고, 다이얼로그를 닫을 수 있도록 app.close() 스크립트를 추가한다
- 다이얼로그를 열었을 때 다이얼로그 내 초점이동이 가능하도록 dialog.focus() 스크립트를 추가한다.

<소스 13. 다이얼로그 닫기 예제(1)>

```
app.openDialog("appURI", { width: 400, height: 300, headerClose: true }, function(dialog) {
    dialog.ready(function(dialogApp) {
        //필요한 경우, 다이얼로그의 앱이 초기화 된 후, 앱 속성을 전달하십시오.
```

```
});
}).then(function(returnValue) {
;
});
```

<소스 14. 다이얼로그 닫기 예제(2)>

```
/*
 * "Button" 버튼에서 click 이벤트 발생 시 호출.
 * 사용자가 컨트롤을 클릭할 때 발생하는 이벤트.
 */
function onClick( /* cpr.events.CMouseEvent */ e) {
    /**
     * @type cpr.controls.Button
     */
    var button = e.control;
    app.close();
}
```

- 다이얼로그 또는 그리드 필터 다이얼로그 열기 시 닫았을 경우, 열기 전 컨트롤에 초점이 이동할 수 있도록 구현한다

<소스 15. 팝업 닫기 후 초점 이동 예제>

```
app.openDialog("appURI", {width : 400, height : 300, headerClose : true}, function(dialog){
    dialog.addEventListener("close", function(e){
        button.focus();
    });
});
```

<소스 16. 그리드 필터 팝업 닫기 후 초점 이동 예제>

```
/*
 * 그리드에서 filterdialog-close 이벤트 발생 시 호출.
 * 필터팝업을 닫을때 발생하는 이벤트.
 */
function onGrdFilterdialogClose( /* cpr.events.CGridEvent */ e) {
    app.lookup("btnOpen").focus();
}
```

[3.2.2 다이얼로그] 예제 참조

6) **tabindex** 속성을 이용하여 키보드 포커스의 순서를 강제로 변경하여 논리적인 순서가 순차적이지 않을 경우, 스크립트 사용으로 인한 비 논리적 이동 및 함정은 오류

[3.2.1 초점 제어] 예제 참조

2.9.2. 초점 또는 키보드의 위치를 나타내는 요소를 시각적으로 표시

1) onFocus="this.blur();" 사용 오류

- focus 이벤트에 blur() API를 사용할 경우 초점이 강제로 사라지기 때문에 해당 스크립트 작성을 하지 않도록 한다

2) hideFocus 적용 시, outline:0, none 적용 시

- CSS를 수정하여 초점의 라인을 없애는 것이 아니라 초점이 시각적으로 표시될 수 있도록 한다
- 컨트롤의 초점 점선라인을 없애는 속성을 사용하지 않도록 한다

<소스 17. outline:0, none 오류 예제>

```
.cl-checkbox.cl-focus .cl-checkbox-icon {
  outline : none;
}
```

4) <a> 요소에 포함된 엘리먼트에 float, display, position 등의 속성이 적용된 경우

- HTML을 작성하여 HTML 스니펫 또는 임베디드 페이지 컨트롤을 이용할 경우에는 <a> 요소에 float, display, position 등의 속성을 적용하지 않는다

2.9.3. 텍스트 등 운용할 수 있는 요소에 초점을 제공

- 이미지 또는 초점을 받을 수 없는 요소에 자바스크립트 이벤트를 사용하여 특정 기능을 수행하도록 제공한 경우 마우스를 사용할 수 없는 환경에서 키보드만으로 해당 요소를 실행하는 것이 불가능하기 때문에 초점을 받을 수 있는 요소 혹은 컨트롤을 사용하여 제공하는 것을 권장한다

[3.2.4 onclick 이벤트 및 onkeydown 이벤트] 예제 참조

2.9.4. 키보드 접근이 가능하여 초점을 제공

1) 이미지 또는 초점을 받을 수 없는 요소에 마우스 이벤트만 적용되어 있는 경우

- 이미지 또는 초점을 받을 수 없는 컨트롤에 마우스 이벤트만 적용했을 경우에는 키보드 접근시에도 동일한 기능을 실행할 수 있도록 제공한다

ex) mouseenter 이벤트 제공 시 focus 이벤트도 함께 제공하거나 mouseleave 이벤트 제공 시 blur 이벤트도 함께 제공한다

[3.2.4 onclick 이벤트 및 onkeydown 이벤트] 예제 참조

2) a 요소에 href 속성이 제공되지 않아 초점이 확인되지 않는 경우

- HTML 스니펫 혹은 임베디드 페이지 컨트롤을 이용하여 <a> 요소를 사용할 때는 반드시 href 속성을 제공하여 키보드 접근 및 초점이 표시될 수 있도록 한다

2.10. 조작 가능

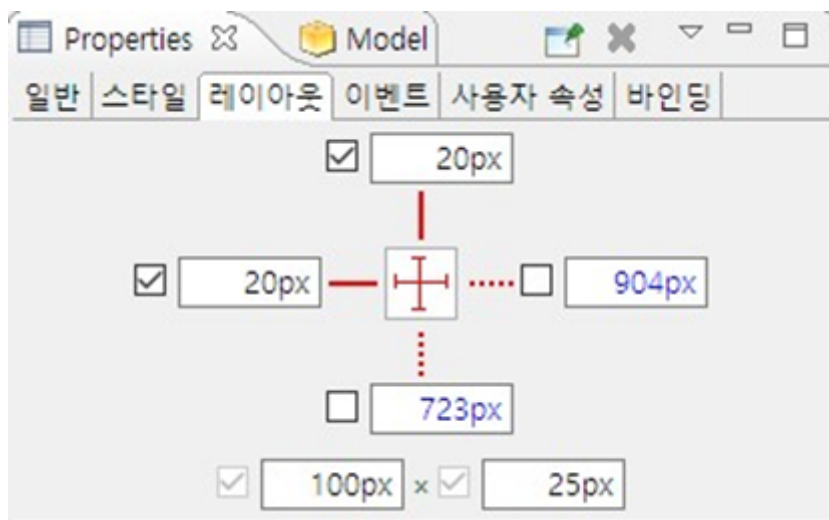
(조작 가능) 사용자 입력 및 컨트롤은 조작 가능하도록 제공되어야 한다.

준수기준 : 웹 페이지에서 제공하는 모든 이웃한 컨트롤들은 개별적으로 선택하고 사용할 수 있도록 대각선 길이가 6.0mm 이상의 크기와 컨트롤의 안쪽 여백이 1 픽셀 이상을 갖도록 제공되어야 한다

2.10.1. 컨트롤의 크기가 대각선으로 6.0mm 이상으로 제공

- 컨트롤의 크기가 대각선으로 6.0mm 미만이 되지 않도록 컨트롤의 크기를 조정한다. 컨트롤 크기 조정은 Properties 뷰의 레이아웃에서 변경할 수 있다

<그림 22. 컨트롤 크기 변경>



- 통상적으로 라디오 버튼 사이즈보다 작은 경우 오류로 분류된다

2.10.2. 링크, 사용자 입력, 기타 컨트롤의 테두리 안쪽으로 1 픽셀 이상의 여백을 제공

- 링크, 사용자 입력, 기타 컨트롤의 테두리 안쪽으로 1 픽셀 이상의 여백을 주어 간섭이 없도록 제공해야 한다

2.11. 응답 시간 조절

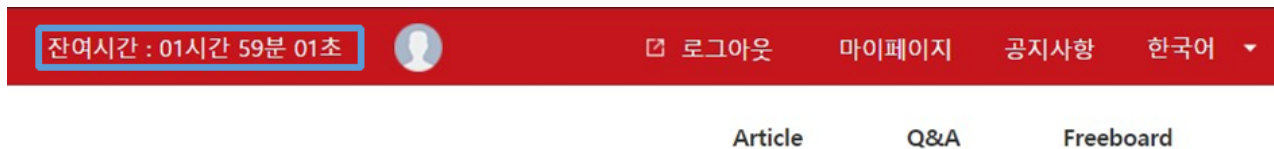
(응답 시간 조절) 시간 제한이 있는 콘텐츠는 응답 시간을 조절할 수 있어야 한다. 준수기준 : 시간 제한이 있는 콘텐츠의 응답 시간을 조절할 수 있어야 한다

2.11.1. 시간 제한이 있는 콘텐츠를 제어할 수 있는 수단을 제공

1) 로그아웃, 인증 요청 등 시간제한이 있는 콘텐츠에 남은 시간을 알려주지 않거나 제어할 수 있는 수단을 제공하지 않은 경우

- 사용자가 로그인 후 제한 시간 또는 남은 시간을 알려줄 수 있도록 로딩 화면을 제공할 수 있다

<그림 23. 로그인 후 잔여 세션시간>



2) 페이지가 전환될 경우 이를 제어할 수 있는 수단을 제공하지 않은 경우

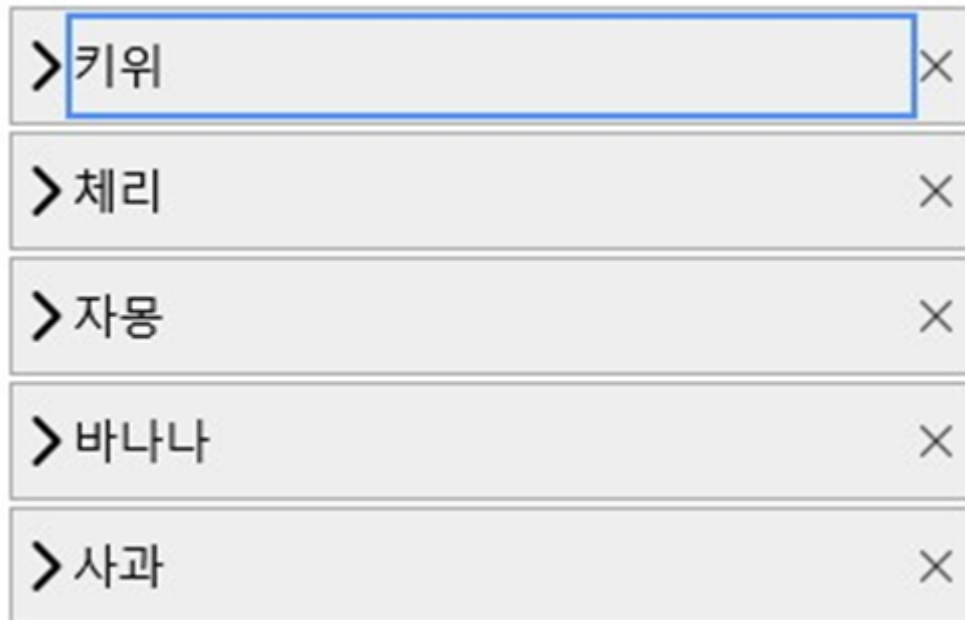
- 페이지 내에 수동으로 갱신할 수 있는 버튼을 배치하거나 해당 페이지가 일정시간 이후 자동으로 갱신된다는 것을 알려 스스로 자동 갱신을 제어할 수 있도록 정지 또는 연장 수단을 제공한다

3) 제한 시간이 있는 콘텐츠의 경우 해당 내용을 확인할 수 있는 수단 제공

- 알림컨트롤은 불특정 하계 시스템이 사용자에게 알림 메시지를 출력할 수 있도록 제공하는 컨트롤이다.
- 알림메시지는 기본(default), 성공(success), 정보(info), 경고(warning), 위험(danger)의 타입을 가지며 정해진 시간(delay) 만큼 보인 후 화면에서 사라진다.
- 일정 시간이 지난 후 사라지는 알림메시지의 경우 사라진 알림 내용을 확인할 수 있는 알림방을 제공하거나 닫기 버튼을 제공하여 사용자가 임의로 메시지를 닫기 전까지 메시지를 화면에 유지시켜야 한다

i 알림방이란, 지난 알림 내용을 리스트 형태로 확인할 수 있는 영역을 뜻한다

<그림 24. 닫기 버튼을 제공하는 알림컨트롤>



- 알림컨트롤의 delay 속성을 충분히 설정하고 close 속성을 true 로 설정하면 각 알림메시지에는 닫기 버튼이 보여지고 닫기 버튼을 클릭하기 전까지 메시지를 유지시킬 수 있다

<소스 18. 닫기 버튼을 제공하는 알림컨트롤>

```
/*
 * Body에서 init 이벤트 발생 시 호출.
 * 앱이 최초 구성될 때 발생하는 이벤트 입니다.
 */
function onBodyInit( /* cpr.events.CEvent */ e) {
    var notifier = new cpr.controls.Notifier("noti");
    notifier.close = true;
    notifier.delay = 9999999999;
    notifier.addEventListener("notify", function(e) {
        var noti = e.control; // 알림메시지가 출력되었을 때 메시지에 포커스 주기
        setTimeout(function() {
            noti.focusItem(e.item);
        }, 1000);
    });
    app.getContainer().addChild(notifier, {
        "top": "20px",
        "right": "20px",
    });
}
```

```

        "width": "300px",
        "height": "30px"
    });
}

```

<그림 25. 알림방제공>



- 일정시간 후 알림메시지가 자동으로 사라지도록 알림컨트롤을 사용하는 경우 위와 같이 전체 알림 메시지를 볼 수 있는 알림방을 제공해야 한다
- 알림방은 알림메시지가 보일 때 메시지 목록을 Array에 저장하고 있다가 알림방 목록 출력 이벤트가 발생했을 때 Array에 있는 내용을 출력하는 방식으로 구현이 가능하다

2.11.2. 제한 시간을 해제 또는 연장하는 방법에 충분한 시간을 제공

- 시간제한에 대한 연장은 연장수단에 도달하기까지 걸리는 부분에 대한 시간의 10 배로 제공해야 한다
- 현행에서는 시간제한에 대한 연장 또는 정지 방법에 대한 안내는 최소 20 초 이상 남아있는 상태에서 제공해야 웹 접근성을 준수한 것으로 본다

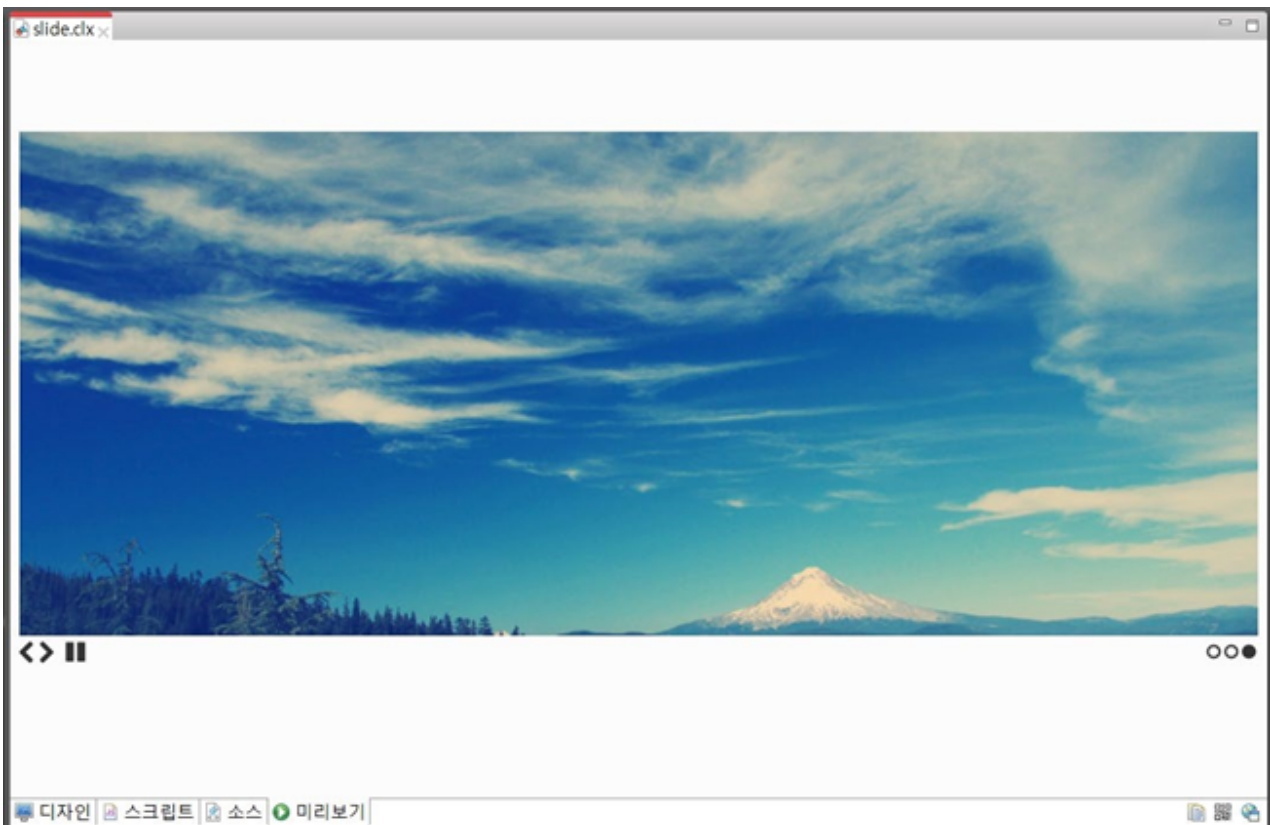
2.12. 정지 기능 제공

(정지 기능 제공) 자동으로 변경되는 콘텐츠는 움직임을 제어할 수 있어야 한다. 준수기준 : 자동으로 변경되는 콘텐츠를 제어할 수 있도록 제공하여야 한다

2.12.1. 시간에 따라 자동으로 변경되는 콘텐츠에 정지, 이전, 다음 기능을 제공

- 자동으로 갱신되는 정보의 경우 정지 기능을 사용하여 충분한 시간을 보내고 정보를 이해할 수 있도록 제공해야 한다

<그림 26. 자동으로 갱신되는 배너>



- tab 키를 통해 이동 또는 재생/정지 버튼에 포커스를 맞출 수 있으며, 엔터키를 눌러 동작할 수 있다

<소스 19. 시간에 따라 자동 변경되는 예시>

```
/*
 * Body에서 load 이벤트 발생 시 호출.
 * 앱이 최초 구성된후 최초 랜더링 직후에 발생하는 이벤트입니다.
 */
function onBodyLoad(e) {
    //일정 시간마다 색을 변경할 수 있도록 타이머를 설정한다.
    setInterval(function() {
        var grp = app.lookup("grp");
        //다음 색상이 없으면 처음 색상으로 표시한다.
    }, 1000);
}
```

```

var ds = app.lookup("ds1");
if (ds.getRowCount() == colorIdx) {
    colorIdx = 0;
}
var color = ds.getValue(colorIdx, "color");
grp.style.css({
    "background-color": color
});
colorIdx++;
}, 3000);
}

```

<소스 20. 마우스 포인터 진입 시 자동 변경 정지 예시>

```

/*
 * 그룹에서 mouseenter 이벤트 발생 시 호출.
 * 마우스 포인터가 컨트롤 위에 진입할 때 발생하는 이벤트.
 */
function onGrpMouseenter(e) {
    var grp = e.control;
    //타이머를 중지한다.
    clearInterval(changeColor);
}

```

2.12.2. 시간에 따라 자동으로 변경되는 콘텐츠를 키보드로 제어 가능

[2.12.1 시간에 따라 자동으로 변경되는 콘텐츠에 정지, 이전, 다음 기능을 제공] 참조

2.13. 깜빡임과 번쩍임 사용 제한

(깜빡임과 번쩍임 사용 제한) 초당 3~50 회 주기로 깜빡이거나 번쩍이는 콘텐츠를 제공하지 않아야 한다.

준수기준 : 초당 3~50 회 주기로 깜빡이거나 번쩍이는 콘텐츠를 제공하지 않았거나, 회피할 수 있는 방법을 사전에 알리고 제공하여야 한다

- 깜빡임의 주기와 크기, 변화의 정도에 따라 사용자에게 피로감 혹은 발작이나 졸도 등을 유발할 수 있기 때문에 해당 콘텐츠는 제공하지 않아야 한다
- 깜빡임 혹은 번쩍임을 중단할 수 있는 수단만 제공하는 것뿐만 아니라 이 사항에 대해 미리 알려 중단 또는 회피할 수 있도록 해야 한다

2.14. 반복 영역 건너뛰기

(반복 영역 건너뛰기) 콘텐츠의 반복되는 영역은 건너뛸 수 있어야 한다. 준수기준 : 반복되는 영역을 건너뛸 수 있는 링크를 제공하여야 한다

2.14.1. 건너뛰기 링크를 제공

- 페이지마다 반복되는 대메뉴를 건너뛰어 본문의 중요 콘텐츠로 이동할 수 있도록 반복영역 건너뛰기 링크를 제공해야 한다
- 네비게이션 바를 통해 직접 원하는 페이지로 이동하는 화면을 구현할 수 있다

[3.4.1 네비게이션바 컨트롤을 이용한 반복영역 건너뛰기] 예제 참조

2.14.2. 건너뛰기 링크의 제공 방법을 적절하게 제공

1) 키보드 운용 시 화면에 보이지 않게 제공된 경우

- 디자인 특성상 건너뛰기 링크를 보이지 않도록 구현하더라도 초점을 받으면 건너뛰기 링크가 표시 되도록 구현한다

[2.8.2 기타] 참조

2) 건너뛰기 링크에 키보드 접근이 불가능한 경우

- 건너뛰기는 키보드 접근이 가능하도록 구현해야 한다
- eXbuilder6 에서의 a 태그는 임베디드 페이지, HTML 스니펫 컨트롤로 구현한다
- HTML에서 a 태그의 focus에 스타일을 준다

<소스 21. 초점 스타일 변경 예시>

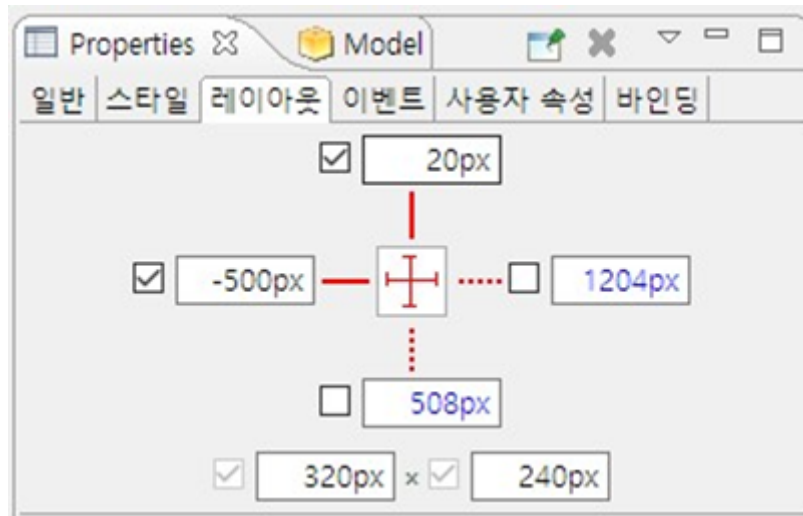
```
a:focus{
  //포커스 CSS 작성
}
```

3) 건너뛰기 링크에 키보드 접근이 불가능한 경우(display:none 또는 visibility:hidden; 속성이 사용된 경우)

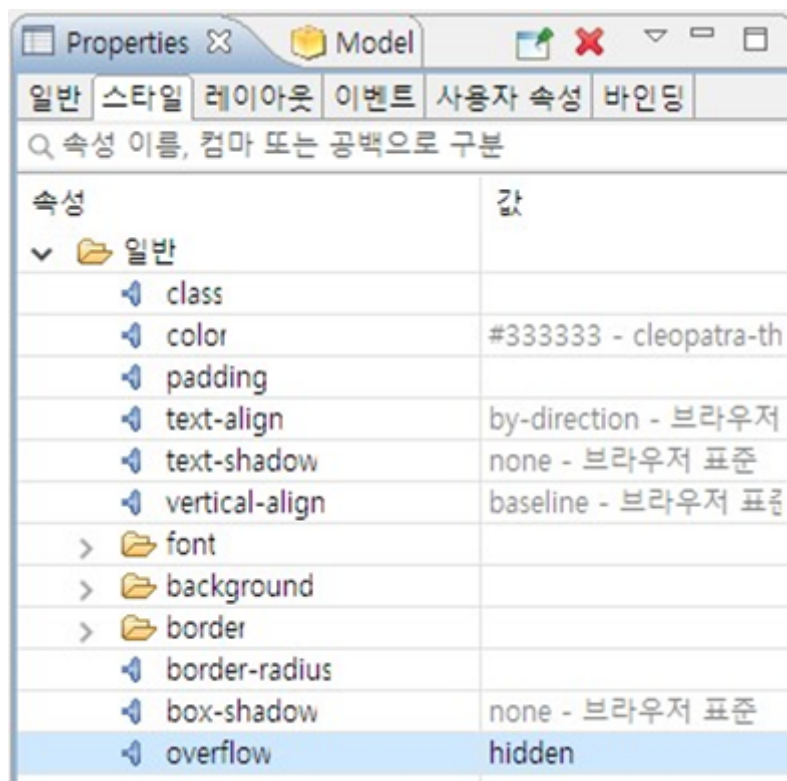
- display:none 혹은 visibility : hidden 속성을 사용하면 키보드에 접근이 불가능하기 때문에 포지셔닝 기법을 사용하여 구현한다

[2.1.1 텍스트 아닌 콘텐츠에 대한 대체 텍스트 제공] 참조

<그림 27. 포지셔닝 기법 사용 - 컨트롤 위치 변경>



<그림 28. 포지셔닝 기법 사용 - overflow: hidden>



4) 건너뛰기 링크를 제공했으나 부적절하게 제공된 경우, 건너뛰기 링크를 제공했으나 href 값 연결 등이 잘못 되어 동작하지 않는 경우

- 링크의 연결 값이 올바르지 않은 경우 해당 기능이 동작하지 않으므로 연결 값을 올바르게 수정한다

5) 건너뛰기 링크를 제공했으나 건너뛰는(도착) 위치가 부적절하게 지정된 경우

- 정상적으로 건너뛰기 링크를 이용할 수 있도록 건너뛰기 링크의 목적지를 올바르게 지정한다

<소스 22. 포커스 이동>

```
app.lookup(ControlId).focus();
```

6) 건너뛰기 링크를 제공하여 화면 스크롤은 되나 실제 키보드 초점은 이동되지 않는 경우

- 건너뛰기 링크는 화면 뿐만 아니라 초점도 함께 다음 영역의 첫번째 항목으로 지정해야 한다
- exbuiler6에서는 포커스가 이동할 경우 자동으로 스크롤도 함께 이동된다

7) 건너뛰기 링크가 페이지의 상단에 제공되지 않은 경우

- 건너뛰기 링크는 반복 영역이 있는 모든 페이지의 상단의 일정한 곳에 제공해야 한다

8) 반복 영역을 건너뛰는 링크 본문 바로가기 등 외에 사용자 운용을 돕기 위한 추가 기능을 제공하였으나 제공 방법이 부적절한 경우

- 모양, 크기, 위치 방향, 색, 소리 등 시각적인 표시에만 그치지 않고 명시적으로 지시 사항을 제공해야 한다

Ex) →(위치방향)만 표시하는 게 아니라 “우측 메뉴 바로가기”와 같은 명확한 지시 사항도 함께 제공해야 한다

[2.4.1 색, 크기, 모양, 방향 등 지시사항을 시각정보 외에 별도 안내 내용과 함께 제공] 참조

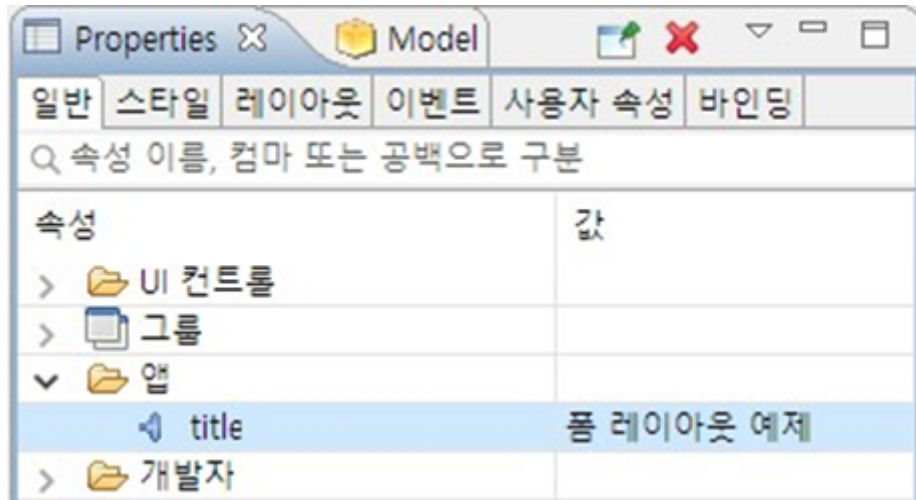
2.15. 제목 제공

(제목 제공) 페이지, 프레임, 콘텐츠 블록에는 적절한 제목을 제공해야 한다. 준수기준 : 페이지, 프레임, 콘텐츠 블록에 적절한 제목을 제공하여야 한다

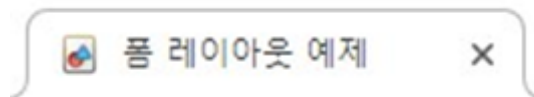
2.15.1. 페이지에 제목을 제공

- 페이지의 제목은 해당 내용을 이해할 수 있도록 적절한 제목을 제공해야 한다
- 페이지의 제목은 App의 title 속성을 변경하여 표시할 수 있다

<그림 29. 페이지에 제목 표시>



<그림 30. 페이지 제목 표시 결과>



2.15.2. 페이지에 제목을 적절하게 제공

1) 페이지 내용과 다른 제목을 제공한 경우, 분류가 더 가능한데 상위 범주로 제목을 제공한 경우, 동일 페이지 제목이 존재하는 경우

- 페이지의 제목을 좀 더 상세하게 분류할 수 있다면 상위 범주를 제목으로 하기보다는 내용을 명확하게 이해할 수 있는 제목으로 제공해야 한다

2) 문서타이틀에 특수문자가 2개 이상 반복된 경우

- 연속된 특수기호 사용시 스크린리더에서 불필요한 음성을 반복해 출력하므로 시각적으로 장식하기 위한 특수문자는 제공하지 않도록 한다

2.15.3. 프레임 요소에 제목을 제공

- 프레임도 요소의 내용을 이해할 수 있도록 제목을 제공해야 한다
- 내용이 없는 프레임(아이프레임)에도 “빈 프레임”, “내용 없음”과 같이 title을 제공해야 한다
- iframe 이 display: none으로 되어 있는 경우 title 속성을 미제공하면 오류로 인식한다
- 임베디드페이지는 내부적으로 iframe을 사용하기 때문에 임베디드페이지를 사용할 때는 반드시 tooltip 속성을 작성하여야 한다

[3.3.2 임베디드 페이지 컨트롤의 타이틀 입력] 참조

2.15.4. 프레임 요소의 제목을 적절하게 제공

- 프레임 요소에 title을 제공할 때는 간결하게 목적이나 용도를 알 수 있도록 해야 한다

Ex) “메인메뉴”, “참고문서” 등으로 설정하는 것은 적절하나 “top 프레임”, “main 프레임” 등은 목적이나 용도를 알 수 없기 때문에 부적절하다

[3.3.2 임베디드 페이지 컨트롤의 타이틀 입력] 참조

2.15.5. 콘텐츠 블록에 제목을 제공

1) 콘텐츠 블록에 h1 ~ h6을 사용하여 제목을 제공하지 않은 경우(콘텐츠 블록의 헤딩이 전혀 제공되지 않는 경우)

- HTML 스니펫 혹은 임베디드 페이지를 통해 h1 ~ h6 요소를 사용할 수 있다

2) 콘텐츠 블록 제목에 display: none 또는 visibility: hidden을 적용한 경우

- 건너뛰기 링크와 동일하게 display: none 또는 visibility: hidden을 사용할 경우 스크린리더에서 접근하지 못하는 문제가 발생하므로 해당 속성을 사용하지 않는다

2.15.6. 콘텐츠 블록에 제목을 적절하게 제공

- HTML 스니펫 혹은 임베디드 페이지를 통해 h1 ~ h6 요소를 사용할 수 있다

1) 콘텐츠 블록 제목에 잘못된 내용을 제공하고 있는 경우

- 콘텐츠 블록 제목은 그 내용을 이해할 수 있도록 제공한다

2) 일관성 있게 <hn>으로 제공되지 않은 경우(<hn>으로 제공된 내용이 중간에 p, span, strong 등 <hn>이 아닌 태그로 제공되는 경우)

- 콘텐츠 블록 제목은 헤딩 요소(h1 ~ h6)를 사용하여 제공해야 한다

3) 콘텐츠 블록 제목의 선형화가 맞지 않을 경우

- 콘텐츠 구조를 이해할 수 있도록 콘텐츠 블록 제목은 논리적인 순서로 선형화가 되어야 한다

Ex) 상위 제목 단계 없이 하위 제목 단계만 있어서는 안 된다

- <hn>의 순서가 잘못된 경우 오류로 볼 수 없어 수정 권장 사항이다

[3.6.1.3 Heading Tag의 활용] 예제 참조

2.16. 적절한 링크 텍스트

(적절한 링크 텍스트) 링크 텍스트는 용도나 목적을 이해할 수 있도록 제공해야 한다.
준수기준 : 링크 텍스트의 용도나 목적을 이해할 수 있도록 제공하여야 한다

2.16.1. 목적이나 용도를 알기 어려운 링크 텍스트를 제공

1) 빈 링크 제공

- 스크린리더를 이용하여 웹 페이지를 탐색하는 사용자의 경우 빈 링크 제공 시 해당 링크의 목적이나 용도를 이해하기 어렵기 때문에 빈 링크로 표시하지 않는다
 - 빈 링크로 표현되지 않도록 하거나 사용하지 않는 링크인 경우에는 삭제한다

2) 빈 링크에 title 속성만 제공한 경우

- 빈 링크 제공과 마찬가지로 빈 링크에 title 속성만 제공할 경우 해당 링크의 목적이나 용도를 이해하기 어렵기 때문에 빈 링크로 표현되지 않도록 하거나 사용하지 않는 링크는 삭제한다.

3) 의미 있는 이미지를 배경으로 제공하고 링크를 빈 값으로 제공한 경우

- 의미 있는 배경이미지에는 IR 기법으로 대체텍스트를 제공해서 링크의 목적을 알 수 있도록 해야 한다

4) 독립적 링크를 포함하는 텍스트 이미지의 alt가 잘못 제공되었거나 빈 값으로 제공된 경우

- 링크의 목적이나 용도를 쉽게 이해할 수 있도록 이미지의 alt 값을 올바르게 제공해야 한다.

[2.1.1 텍스트 아닌 콘텐츠에 대한 대체 텍스트 제공] 참조

5) 링크 내 이미지가 의미를 가지고 있으나 대체 텍스트를 제공하지 않은 경우(alt 미제공)

- 링크의 목적이나 용도를 쉽게 이해할 수 있도록 이미지의 alt 값을 올바르게 제공해야 한다

[2.1.1 텍스트 아닌 콘텐츠에 대한 대체 텍스트 제공] 참조

6) 목적이나 용도를 알기 어려운 링크 텍스트를 제공한 경우

- 목적이나 용도를 이해하기 어려운 단순 동작의 내용만을 담은 링크 텍스트로 제공하면 직관적으로 해당 파일의 성격이 무엇인지 이해하기 어렵기 때문에 title 속성을 지정하여 마우스 사용자는 툴팁으로, 스크린리더 사용자는 title을 읽어 좀 더 명확하게 파일의 성격을 구분할 수 있도록 제공해야 한다
- title 속성은 exbuilder6의 각 컨트롤의 tooltip 속성으로 제어가 가능하다

7) 링크 텍스트를 단순히 URL 경로로만 제공한 경우

- URL 경로에 링크를 걸지 않고 “메일보내기”와 같은 버튼으로 제공하는 것을 권장한다

8) 링크의 정보를 이미지의 요소에서 알리는 경우

- 링크 정보는 이미지 요소에 제공하지 않고 a 요소에 title로 정보를 제공하는 것을 권장한다

Ex) `<a href="URL"></a>`보다는 `<a href="URL" title="새창열림"></a>`

9) “~를 하려면 여기를 누르세요”와 같이 방향지시로 링크 목적을 안내한 경우

- 어떠한 기능을 하는 링크인지는 클릭을 해서 해당 링크 페이지로 이동하기 전까지는 알 수 없기 때문에 링크텍스트는 직관적이고 의미 있게 구성해야 한다
 - 링크텍스트는 사용자가 링크의 용도나 목적을 명확히 알 수 있도록 수정한다

2.17. 기본 언어 표시

(기본 언어 표시) 주로 상아는 언어를 명시해야 한다.

준수기준 : 기본 언어 표시 속성을 사용하여 언어를 명시하여야 한다

2.17.1. 문서 타입에 맞는 기본 언어를 명시

- HTML 문서의 lang 속성을 정의하여 기본 언어의 표현은 exbuilder6 설정파일을 통해 정의가 가능하다

[3.5.1 exbuilder.json 의 html-lang 속성을 이용한 기본 언어 설정] 예제 참조

2.17.2. 문서 타입에 맞는 기본 언어를 적절하게 명시

- 각 컨트롤의 lang 속성은 미지원

2.18. 사용자 요구에 따른 실행

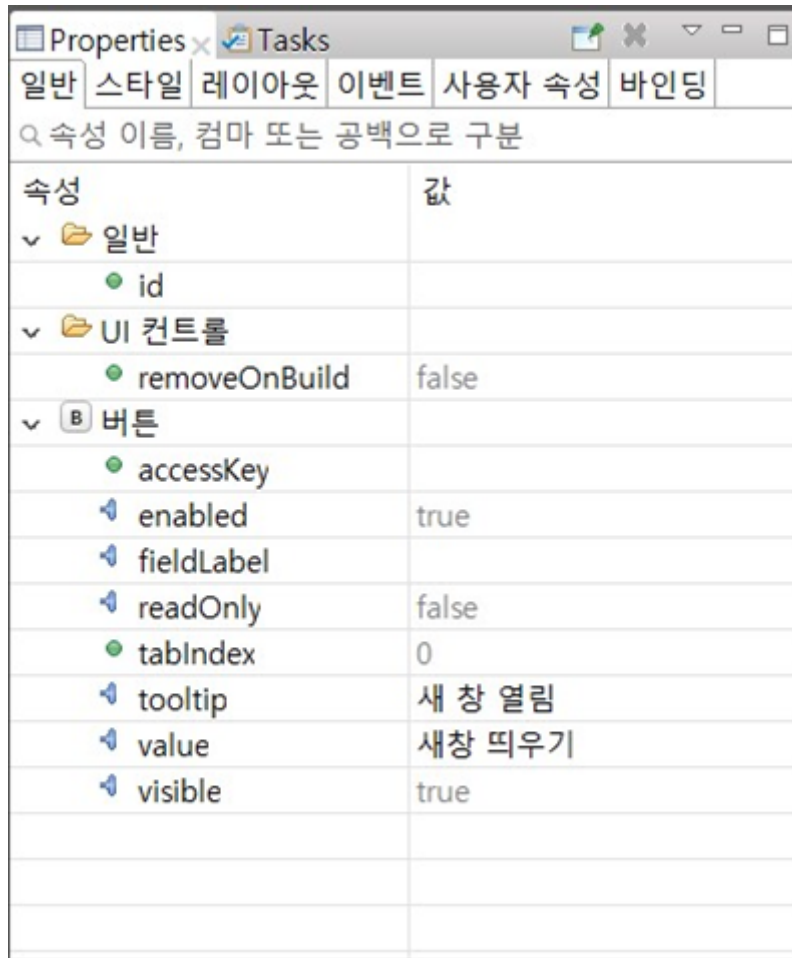
(사용자 요구에 따른 실행) 사용자가 의도하지 않은 기능(새 창, 초점 변화 등)은 실행되지 않아야 한다.

준수기준 : 사용자가 의도하지 않은 기능이 실행되지 않도록 해야 한다

2.18.1. 사용자가 실행하지 않은 상황에서 예측하지 않은 새 창이 열리는 경우

- tooltip 속성을 이용하여 새 창이 열릴 것이라고 미리 정보를 제공할 수 있다

<그림 31. tooltip 속성 수정>



페이지 시작 시, 새 창 또는 화면을 가리는 레이어 팝업을 제공

- 페이지 로딩 시 제공되는 화면을 가리는 레이어 팝업 또는 모달창은 반복 영역 건너뛰기 보다 앞서서 초점이 접근될 수 있도록 구현되어야 한다

[3.2.2 다이얼로그] 예제 참조

2.18.3. 사용자가 의도하지 않은 초점 변화나 기능 동작으로 불편을 주는 경우

1) 초점이 갑자기 사라지지 않도록 지원

- `onfocus = blur();` 기능을 제거함으로써 초점을 잃지 않도록 할 수 있다

<소스 23. blur() 기능 제거필요>

```
<a href="#" onfocus="blur();>test 링크</a>
```

2) 팝업창에서 ‘다시 보지 않기’ 등의 확인란 선택으로 자동 서식이 제출되어, 팝업이 사라지는 경우

- 확인, 닫기 버튼 등을 제공하여 갑자기 팝업이 사라지는 것을 사용자가 인지할 수 있도록 한다

[2.18.2 웹 페이지 시작 시, 새 창 또는 화면을 가리는 레이어 팝업을 제공] 참조

3) 요소박스에서 데이터 입력 시, 다음 영역으로 자동 이동하는 경우

- input 태그에서 입력을 완료했을 경우, keyup 이벤트를 삭제함으로써 다음 영역에 자동으로 포커스가 이동하지 않도록 수정할 수 있다

4) Dropdownlistbox에서 키보드로 옵션 선택 시, 링크가 이동되는 경우

- 선택한 옵션으로 이동하는 버튼을 추가하여 옵션 선택 후 버튼 클릭을 통해 이동할 수 있도록 지원할 수 있다

[3.2.3 컨트롤의 내부 이동 단축키] 예제 참조

<그림 32. 선택한 옵션으로 링크를 이동할 수 있는 버튼 추가>



2.19. 콘텐츠의 선형화

(콘텐츠의 선형 구조) 콘텐츠는 논리적인 순서로 제공해야 한다. 준수기준 : 콘텐츠의 순서가 논리적으로 선형화 되어 제공되어야 한다

2.19.1. 콘텐츠의 순서가 논리적으로 제공되어 계층 구조를 이해할 수 있도록 제공

1) [제목-내용] 콘텐츠 목록의 배치가 분리되어 이해가 불가능한 경우

- 각 탭 페이지 상단에 표현하는 페이지의 타이틀을 제공하여 이해를 쉽게 할 수 있다

<그림 33. 탭 폴더, MDI 폴더의 탭 아이템 이름 지정>

원그리드 조회 X 원그리드 페이지 X

All Other Toggle CHK

원그리드 페이지

column1 column2 column3 column4 column5 column6 column7 column8 column9

출건수: 200 페이지: 1 / 7

	column1	column2	column3	column4	column5	column6	column7	column8	column9	c
1	pLyG+hVA430tg=	GxtcS4b0EvSkvw6azRUBS4U9Uw=	gRW3p6Afw3UA=	mhBqZk/kojKanO	rw6vSnBYaDtqeA	6GT/yE6ofxPO=	BOEkT8/y9/eNpgz	Zm90J/eyym7w	NQc	
2	qJ7cGAGCkm32rAxdLFelgcauhF3Q	PMFC+gJa7Be50	Bg0kMiBeyPiwZ	QwOCyy5e3Si5wdvtag	C4wCdWGG4JW5Y37s	YwX6TxAuN03S	+PKVC+u5AHZ	4TKdL6Ztcxow	wywx	
3	rkVeBxF7uZcvA=	Holynugi8P0pZA=	QYnPe1QyJFA3Q	cRet9a99nq4C	OwPiWhMZEOkkmG8H	/geCcRiBg2Fbg	ba6oBvo0vWjN8A	UlcCNwb9bHJbiQ	52/4r/P02+pk0Q	=5347
4	ucodqsgo9TcG+g	mqGU+FP64kWNE	YxAKtOsM2s7Jp	CtqNoVN9/flicdA	=cXhInNNUU/N9T	wws4YEHY6YN18	//9NGRxd/XkUA	=hca+XQF7lkd	dhQ+rkzXku1IT	+NLnw=GTJ2
5	aV6ot1ZSIFRCa	gCgm+AXu40csR	CNUi5+9akGA5J	+wMjZprRIOGLU	73QmqXvFFDQicoo	QwBsVESuXTaD	J3le2DSqHmwd	WUue+g9puNQC	SvmvGL15dWV	5cmmx6Qd2nT
6	ezbnGqSlnVEOB	gVLOVWQRJEJah	BcqhVvhobPbqq	CgkSe/ok9a/Zlog	=WZJsgSSxGSHg	UCC6CwvxdMqSH	/0zuV2/bi8YvHw	=qsTgusc3XATI	QOPzCyQD/	HH6egny4yH
7	1Qrb/yzeRRScv	QW0FBa5d50Sv	Vqg2XMYZyAMV	BTFrVIGoVpaGyc	ZcwCMU5Bw4sP	wixtEBTVce5vbr	RnPAFJ6phRC	xXGisUgIv	epQwsOXn+ZG	JDW76XksyZlkKA
8	aulLYZkdwTKz	wqS3HIZ3/1ADk	A=7qXzm6GHKN	wG1APRie4e2ut4	Xw=7MGkFyWNRqu	MVFDTL5Ugzw	FsIHwV3GnMdy	4oraZogxHckRu	YdYzy54QUA	+Z63UU6RAJ8A
9	wN3rZQkNIRpu	gM52CdsAww+so	UTDmWIOIO+/6	CvgJy90a9Cet	WJQdTZS9t73Go	EJYgUnVtmWQ	/MoYfn(SLQhe	BOP8LFzT9u	1D6IM4wp4zd	wZDss9+005V
10	CKwJxygLFwd	lbgccG+BnO6iOf	sHQHkSOIPFW4G	A1K(RKE4oKP	Aluk6QA/xRHpi	Z+3umFDwDBir	7InoxC6sJQ	=ELJED11Pq	XLXGCO0Un7ay	Ku/b/9wihhLEuc
11	A46/PRHorZ7R	3gYz9YsJRS+R	FtJAmsITBcTi	+3hMdA/S8+UG	akWsvutg=0Wx	ArJfNgwscog	9wjqbSCRGA7	//ghPaekz0Fr	+daQ+hbZs2R	5465KGRgave
12	LGg/ZOuHyK/qi	wYLSRucIFApds	rA=BT7Lh76jB	4wLYQ=w8KOQLN	/9NWA9cVIHb	RaglpstQ=	x6PqYC/1Q9R	Sdird55wv4v7	hwYfg=YLxmev	BXU+lvwgR
13	5ZYCASJOL/ap	Dvzefolfd+bl	gsw==IvBdqMIB	yyS1yA=MTHm	U/AXhYzyuunq	NNcSsjPK5Q	=+jmhw18zNG	daJAYM0eTT	4a0yKU7wUH	Jnt0r10Iye1w
14	Jrhze/GmbEY	+Ak30bVgG2O	FhCkvydz6Sd	/Yg==CrdYnYfm	NymOQch37	+Fnk43HJxQ	=mC7mv/107xu	Ug=JMYwFG3P	TjXnOxwPfi2	Phyc/00+te
15	zJ7Pig9z1AF	MBQBUemoXwn	7RbBYdlwW9r	izAkdg==/3G	VVYUv52vrg	=Y2+6spzR/k	23ew/4flockgh	KN8LGA=Q1N	6lom9e37hNg	+CsdHiruNY0
										HriQ=EV23ER
										KckY7qbeuAa

Export 저장

2) 기능적인(로그인, 검색 등) 항목의 순서가 논리적이지 않게 제공한 경우

- 스튜디오의 Outline View를 통해 컨트롤의 논리적인 순서를 조절할 수 있고, 이 순서는 포커스의 이동 순서와 동일하게 제공된다

[3.2.1 초점 제어] 예제 참조

3) 사용방법 안내 정보를 콘텐츠 이전에 제공하지 않은 경우

- 아웃풋 컨트롤을 사용하여 안내 정보를 작성하여 제공할 수 있다.

2.20. 표의 구성

(표의 구성) 표는 이해하기 쉽게 구성해야 한다.

준수기준 : 표를 이해할 수 있도록 정보를 제공한 경우 준수한 것으로 인정

- 그리드 컨트롤을 table 태그가 아닌 div 태그로 제공되어 caption과 summary를 별도로 지원하지 않음

- 캘린더 컨트롤은 내부적으로 table 태그를 사용하며 table 태그를 사용하지 않는 방법으로 적용될 예정입니다

2.21. 레이블 제공

(레이블 제공) 입력 서식에는 대응하는 레이블을 제공해야 한다. 준수기준 : 입력 서식에 대응하는 레이블을 제공하여야 한다.

- 레이블 제공 시 기능정보를 추가제공하는 것이 권장된다.
- eXbuilder6의 컨트롤은 fieldLabel 속성을 제공하고 이 속성은 aria-label 로 표현되어 레이블 제공수단으로 사용된다.

2.21.1. 그리드

그리드에서 제공하는 레이블은 헤더 셀의 text와 그리드 셀value이 함께 제공되어야 한다.

따라서 그리드 셀에 배치되어 있는 컨트롤의 fieldLabel 속성은 헤더 셀의text를 작성한다.

[3.3.1 입력 계열 컨트롤의 레이블 활용 가이드] 예제 참조

2.22. 오류 정정

(오류 정정) 입력 오류를 정정할 수 있는 방법을 제공해야 한다. 준수기준 : 입력 오류를 정정할 수 있는 방법을 제공하여야 한다

- 오류 발생 시, 아웃풋 컨트롤에 발생원인에 대한 내용을 제공하고 해당 입력 서식으로 초점을 맞춰 오류를 정정할 수 있도록 제공할 수 있다

<그림 34. 오류 메시지와 오류가 발생한 서식에 맞춰진 초점>

The image shows a web form for login. It includes a text input field at the top, followed by a password input field with the label '비밀번호'. Below the password field is a checkbox with the label '아이디 저장'. A red error message '아이디를 입력해주세요' is displayed below the checkbox. To the right of the input fields is a '로그인' button.

- 오류 발생 시, 입력한 내용을 삭제하지 않고, 오류가 발생한 지점에 초점을 맞추어 오류만 수정할 수 있도록 제공할 수 있다

- 포커스 아웃 시 입력된 내용이 부적절하다고 하여도 내용을 초기화하지 않도록 하거나 사전에 입력 가능한 형식에 대한 정보를 제공하여야 한다. 이 정보는 입력 컨트롤의 tooltip 속성에 설정하여 제공이 가능하다.

<그림 35. 입력 가능한 형식 정보 제공 예시>



2.23. 마크업 오류 방지

(마크업 오류 방지) 마크업 언어의 요소는 열고 닫음, 중첩 관계 및 속성 선언에 오류가 없어야 한다.

준수기준 : 마크업 언어 요소의 열고 닫음, 중첩 관계 및 속성 선언에 오류 없이 제공하여야 한다

- HTML 스니펫 컨트롤과 HTML 페이지 컨트롤을 제외한 모든 컨트롤의 마크업 오류는 툴에서 보증한다

[3.6.1 HTML 스니펫 컨트롤의 활용] 예제 참조

2.23.1. 태그의 열고 닫음 오류

- HTML 오브젝트 또는 HTML 스니펫 컨트롤을 사용하여 작성된 HTML 태그는 시작 태그와 종료 태그 형식을 맞추어 제공되어야 한다

<소스 24. div 태그의 열고 닫음>

```
<div></div>
```

2.23.2. 태그의 중첩 오류

- HTML 오브젝트 또는 HTML 스니펫 컨트롤을 사용하여 시작 태그와 종료 태그의 위치가 다른 태그와 중첩되지 않도록 제공되어야 한다

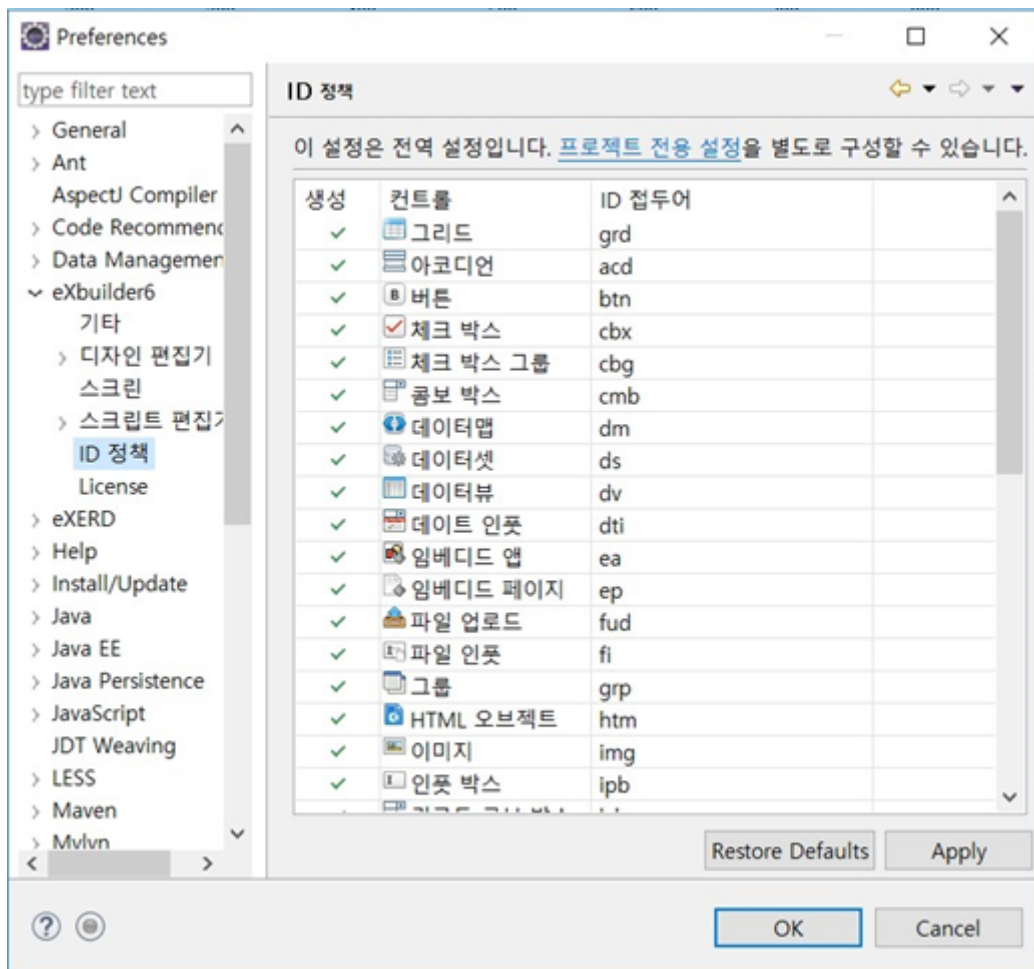
<소스 25. 태그의 포함관계>

```
<div>
  <h2>웹 접근성 준수</h2>
</div>
```

2.23.3. 중복 선언된 속성 오류, id 값 중복 사용 오류

- Window > Preferences > eXbuilder6 > ID 정책 을 통해 컨트롤의 ID 접두어를 설정할 수 있다.
- 설정 후 Id 속성을 지정 시, 중복이 되지 않도록 제공할 수 있다
- 스튜디오에서 각 컨트롤 별 속성 뷰에서 설정된 Id 는 App 의 범위에서 관리하는 논리적인 ID 로 app.lookup 등의 메소드를 통해 접근할 수 있고 실제 HTML 로 그려질 때는 전체 HTML 범위에서 ID 가 중복되지 않도록 설정된 Id 와 매핑되는 uuid 가 생성되어 설정되다
- 각 컨트롤의 ID는 App 내에서 유일성이 보장되도록 설정한다

<그림 36. eXbuilder6 의 ID 정책>



2.23.4. 웹 표준 문법에 맞지 않는 경우

- 임베디드 페이지, HTML 오브젝트 또는 HTML 스니펫 컨트롤을 사용하여 임의의 HTML 태그를 삽입할 경우 삽입되는 태그는 웹 표준 문법에 맞게 작성되어야 한다
- 사용자에게 의해 만들어진 임의 태그를 삽입하는 경우가 아닌 eXbuilder6에 의해 자동으로 만들어지는 모든 태그는 웹 표준 문법에 맞도록 틀에서 보장한다

2.24. 웹 애플리케이션 자체 접근성

(웹 애플리케이션 접근성 준수) 콘텐츠에 포함된 웹 애플리케이션은 접근성이 있어야 한다. 준수기준 : 웹 애플리케이션의 자체 접근성을 준수하였거나 대체 콘텐츠를 제공한 경우 준수한 것으로 인정

- 접근성이 어려운 웹 애플리케이션(플래시, 기타 모듈 등)의 콘텐츠는 자체적인 접근성을 확보할 수 있는 HTML 버전으로 제공한다
- “HTML 오브젝트” 컨트롤을 이용하여 PDF 뷰어 등과 같은 콘텐츠 뷰어 기능을 제공할 때는 PDF 문서를 다운로드 받을 수 있는 링크를 별도로 제공해야 한다. 또한 제공되는 문서는 스크린리더 사용자가 PDF를 읽을 수 있도록 이미지 기반 문서(스캔 등을 이용하여 이미지 파일로 제작한 문서)가 아닌 텍스트 기반 문서로 제공되어야 한다



문서가 텍스트 기반임을 확인할 수 있는 가장 쉬운 방법은 텍스트를 마우스로 드래그하여 텍스트 선택이 가능한지 확인하고 이렇게 선택된 텍스트를 복사/붙여넣기 하여 텍스트 편집기로 붙여 넣었을 때 온전한 텍스트가 붙여 넣기 되는 것을 확인하는 것이다.

- 스크린리더 중 “센스리더”를 사용 시, 그리드 컨트롤을 위해 가상 커서를 해제하여 이용하라는 안내 문구를 제공해야 한다. 가상 커서를 해제함으로써 그리드의 필수적인 정보만 제공받을 수 있다

3. 적용 예제

3.1. 대체 텍스트 처리

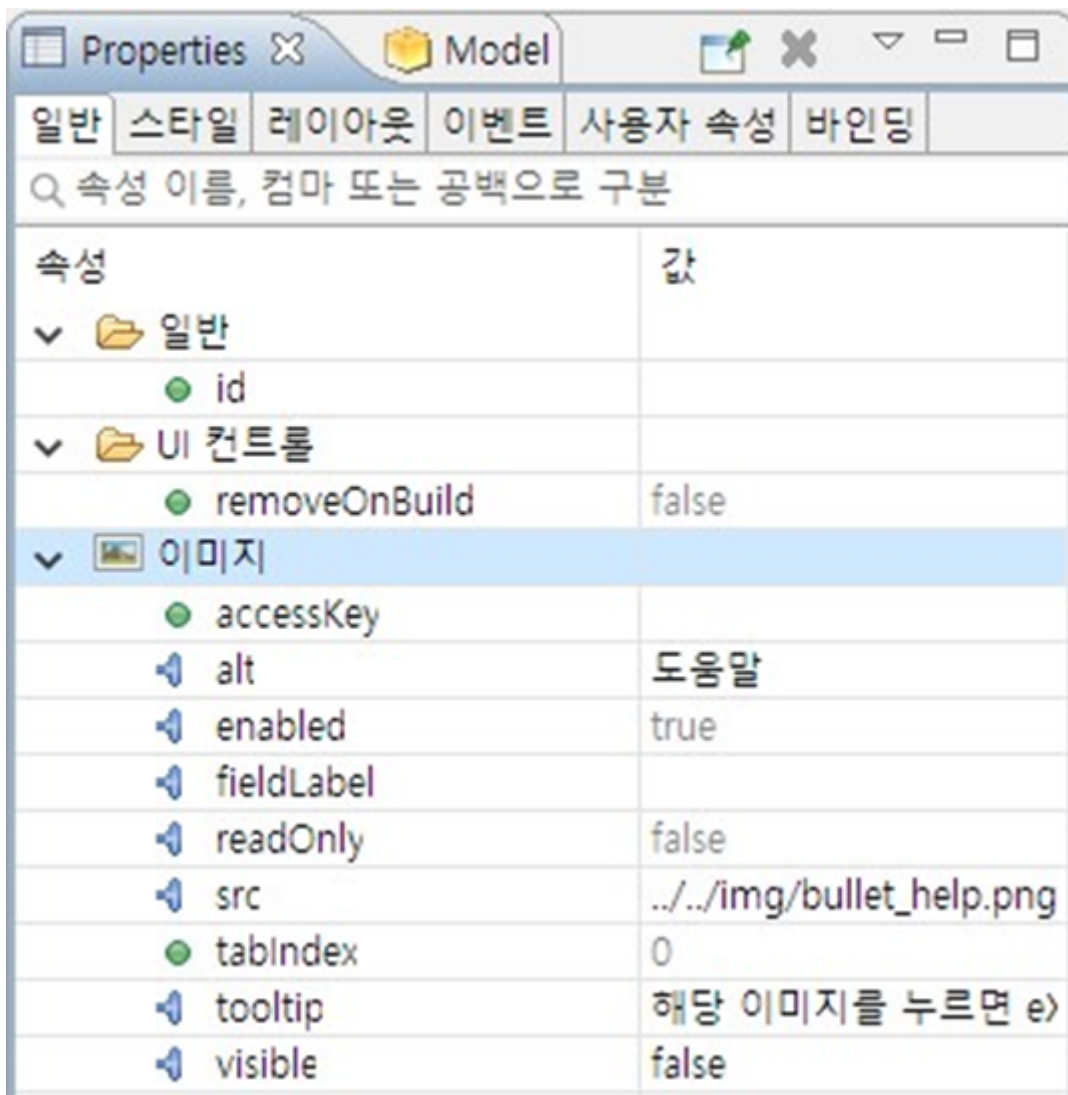
3.1.1. 이미지 컨트롤

- 이미지 컨트롤을 디자인 탭에 배치한 후 대체 텍스트를 설정한다

3.1.1.1. 대체 텍스트 처리

- 이미지 컨트롤을 배치한 후 Properties 탭의 alt 속성과 tooltip 속성을 지정하여 대체 텍스트를 처리할 수 있다(alt는 이미지에 대한 설명이며 tooltip 과 중복된 내용은 피해야 한다)
- 대체 텍스트의 설명이 부족한 경우 속성을 더블 클릭하여 수정할 수 있다

<그림 37. 이미지 컨트롤 alt 속성 설정>



위의 속성을 입력하면 다음과 같은 결과가 나타난다.

<소스 26. alt 속성과 tooltip 속성 적용 결과>

```
<img style="width:100%; height:100%; display:block"
  data-ndid="f817e1ee-a70d-f7c7-d589-f1960b5cab9c" alt="도움말"
  title="해당 이미지를 누르면 eXbuilder6에 대한 도움말을 확인할 수 있습니다."/>
```

eXbuilder6의 콘텐츠는 도움말 구성, 소개, 개발환경, 라이선스, 튜토리얼, 스튜디오 구성, 프로젝트, 앱 개발, 고급기능, 외부연동, API Reference의 내용을 살펴볼 수 있습니다."

src="theme/images/tomatosystem.png">

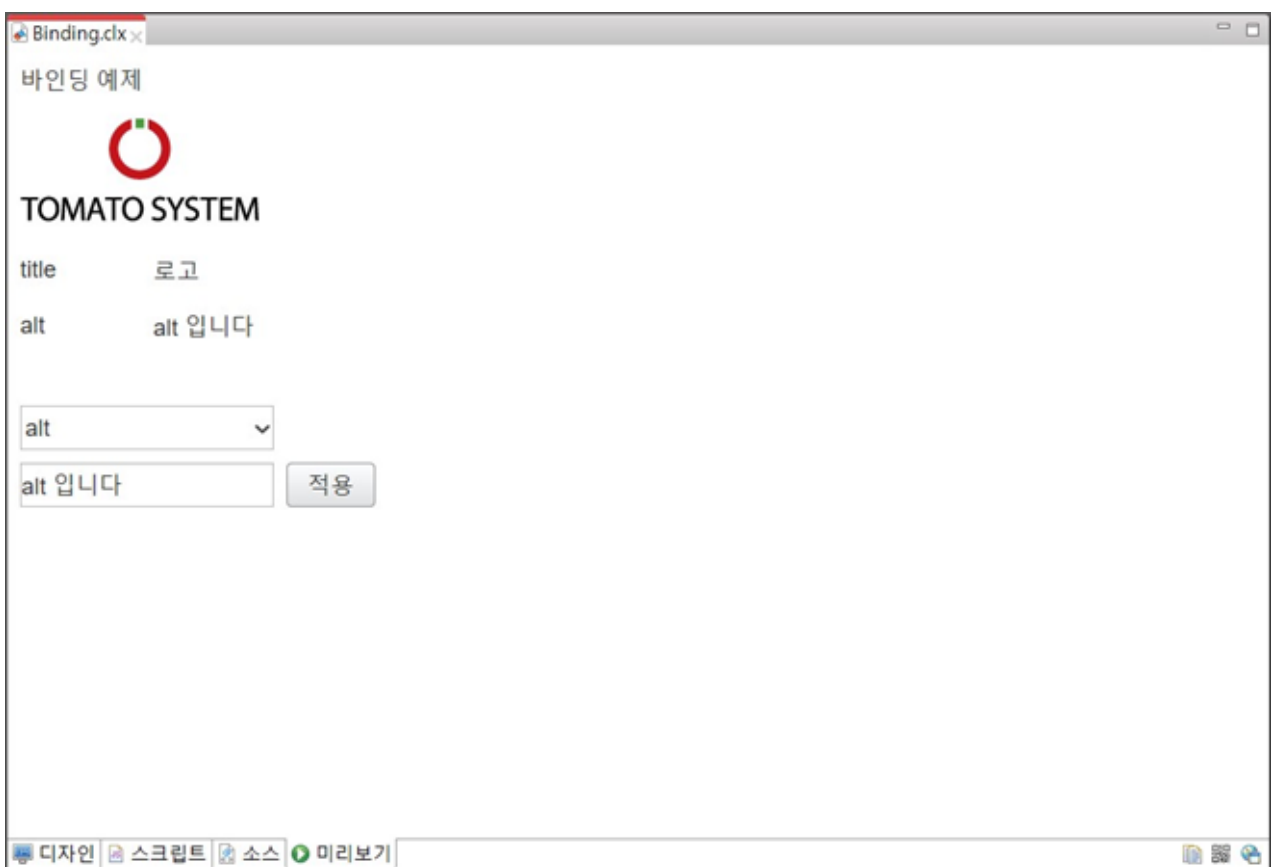
- tooltip 속성으로 제공한 대체텍스트는 브라우저에서 title 속성으로 확인할 수 있다

관련항목: 2.1. 적절한 대체 텍스트 제공 - 1, 2, 4

3.1.1.2. 바인딩

- 사용자가 대체 텍스트 등록을 할 수 있도록 alt 와 tooltip 속성을 바인딩 할 수 있다
- 콤보 박스 컨트롤에서 변경할 속성을 선택한 뒤, 인풋 박스 컨트롤에 텍스트를 입력한다
 - 작성한 후, '적용' 버튼을 누르면 스크립트를 통해 대체 텍스트의 내용을 수정할 수 있다

<그림 38. 데이터 바인딩 화면>



<소스 27. 사용자가 직접 alt, title 속성 설정>

```

/*
 * "적용" 버튼에서 click 이벤트 발생 시 호출.
 * 사용자가 컨트롤을 클릭할 때 발생하는 이벤트.
 */
function onBtn1Click( /* cpr.events.CMouseEvent */ e) {
    /**
     * @type cpr.controls.Button
     */
    var btn1 = e.control;
    var img = app.lookup("logo");
    var val = app.lookup("cmb").value;
    var inpt = app.lookup("ipb").value;

    if (val == "title") {
        img.tooltip = inpt;
    } else if (val == "alt") {
        img.alt = inpt;
    }

    app.lookup("ipb1").value = "";
    app.lookup("grp1").redraw();
}

```

관련 항목 : 2.1. 적절한 대체 텍스트 제공 - 3

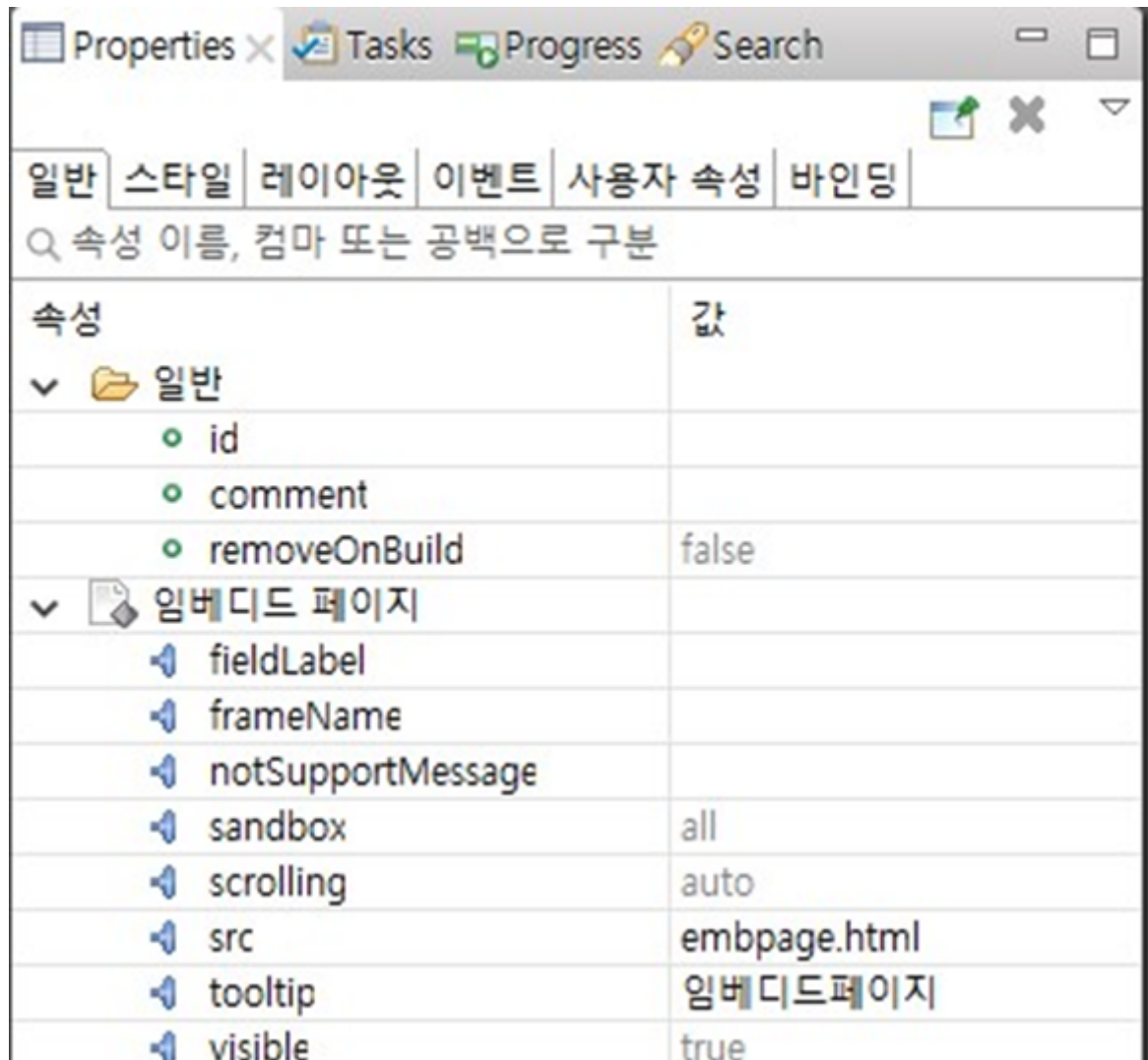
3.1.2. 임베디드페이지

- 임베디드페이지 컨트롤을 디자인 탭에 배치한 후 대체 텍스트를 설정한다

3.1.2.1. 대체 텍스트 처리

- 임베디드페이지에 표시되는 iframe에 title 속성을 설정하여 표시한다

<그림 39. 임베디드페이지 tooltip 속성 설정>



관련 항목: 2.1. 적절한 대체 텍스트 제공 - 1, 2, 4

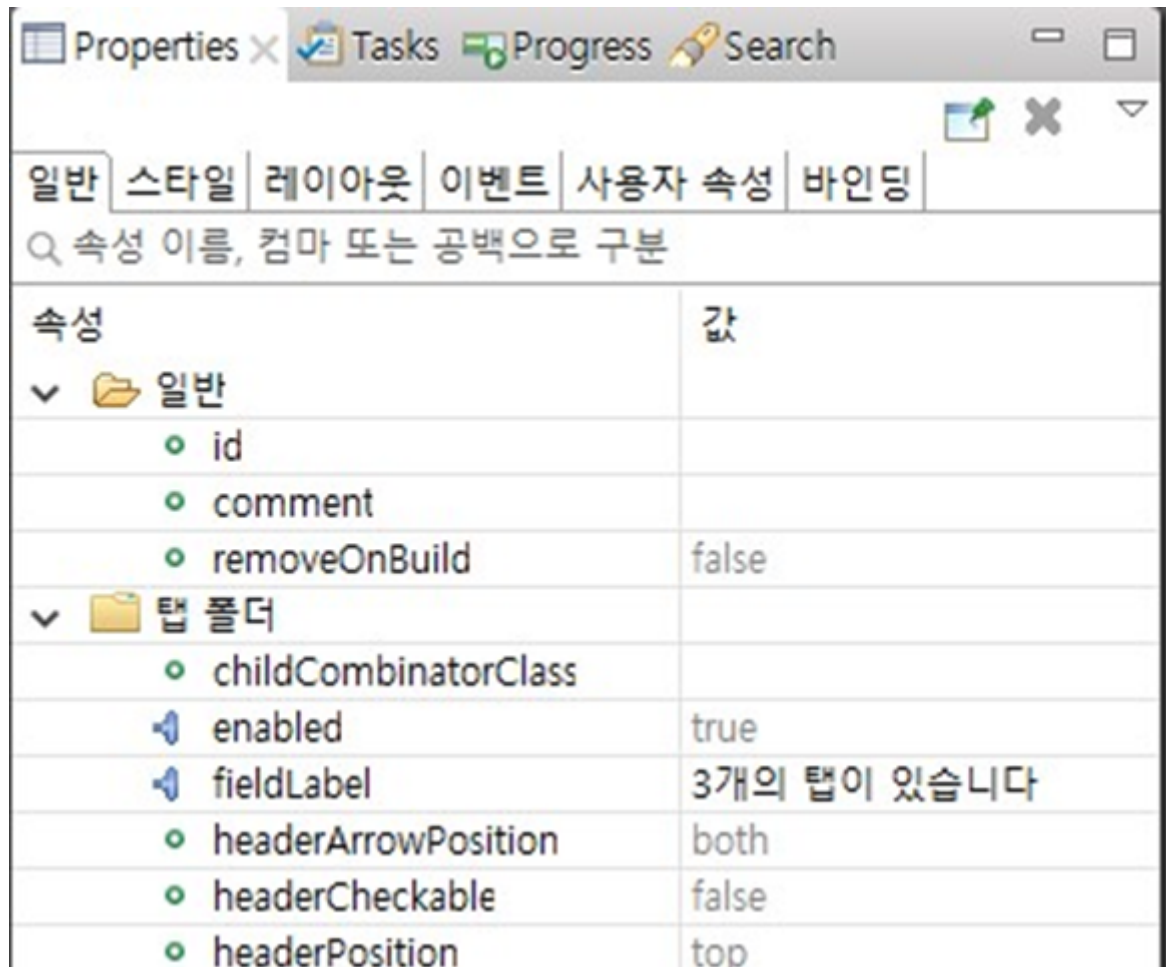
3.1.3. 탭폴더/MDI 폴더

- 탭폴더 컨트롤을 디자인 탭에 배치한 후 대체 텍스트를 설정한다

3.1.3.1. 대체 텍스트 처리

- fieldLabel 속성에 값을 설정하여 스크린리더에서 읽을 수 있게 한다

<그림 40. 탭폴더 fieldLabel 속성 설정>

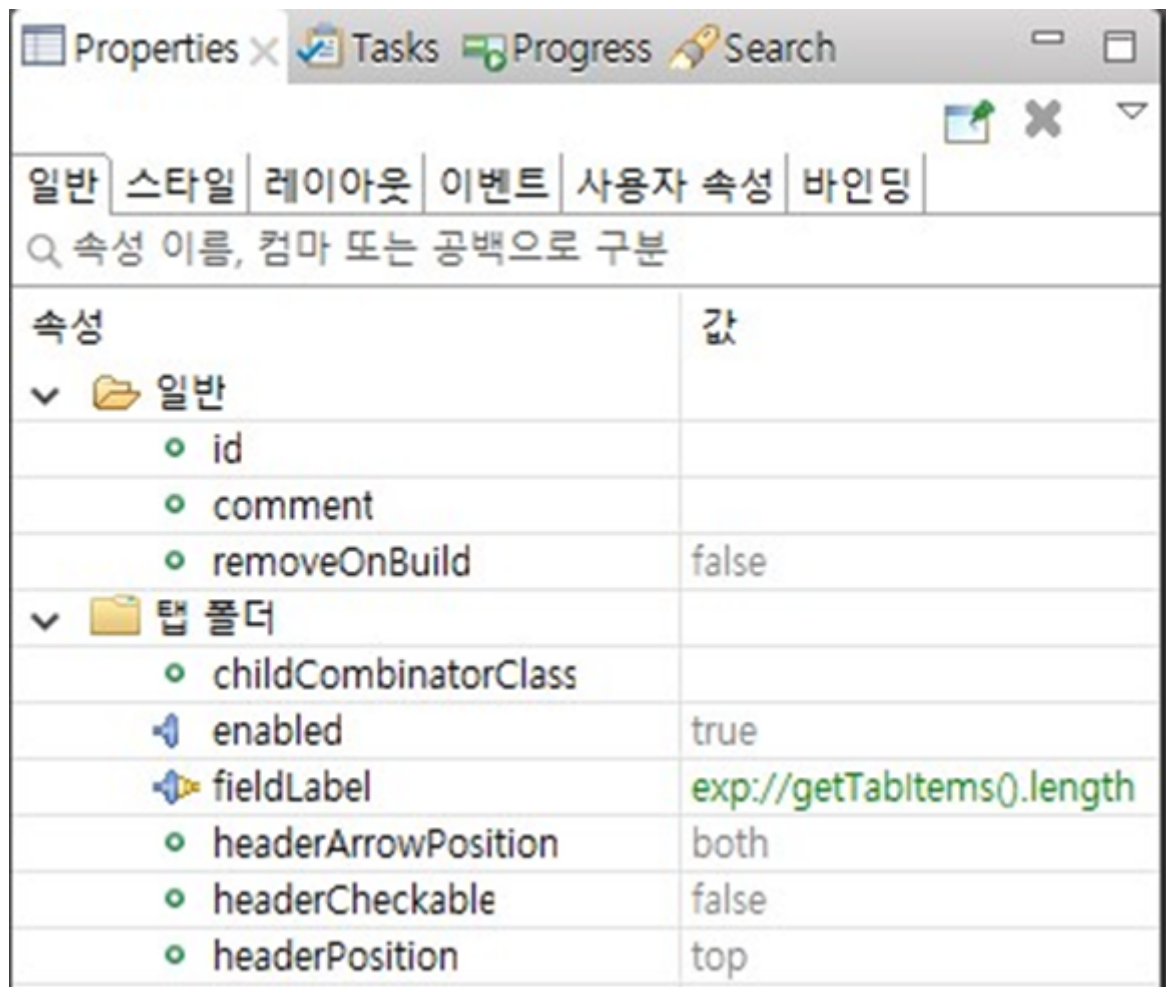


관련 항목: 2.1. 적절한 대체 텍스트 제공 - 1, 2, 4

3.1.3.2. 바인딩

- 익스프레션 바인딩을 통해 동적으로 탭의 개수를 읽을 수 있게 한다

<그림 41. 탭폴더의 fieldLabel 익스프레션 설정>



<소스 28. 익스프레스션 설정 값>

```
getTabItems().length + '개의 탭이 있습니다.'
```

관련 항목 : 2.1. 적절한 대체 텍스트 제공 - 3

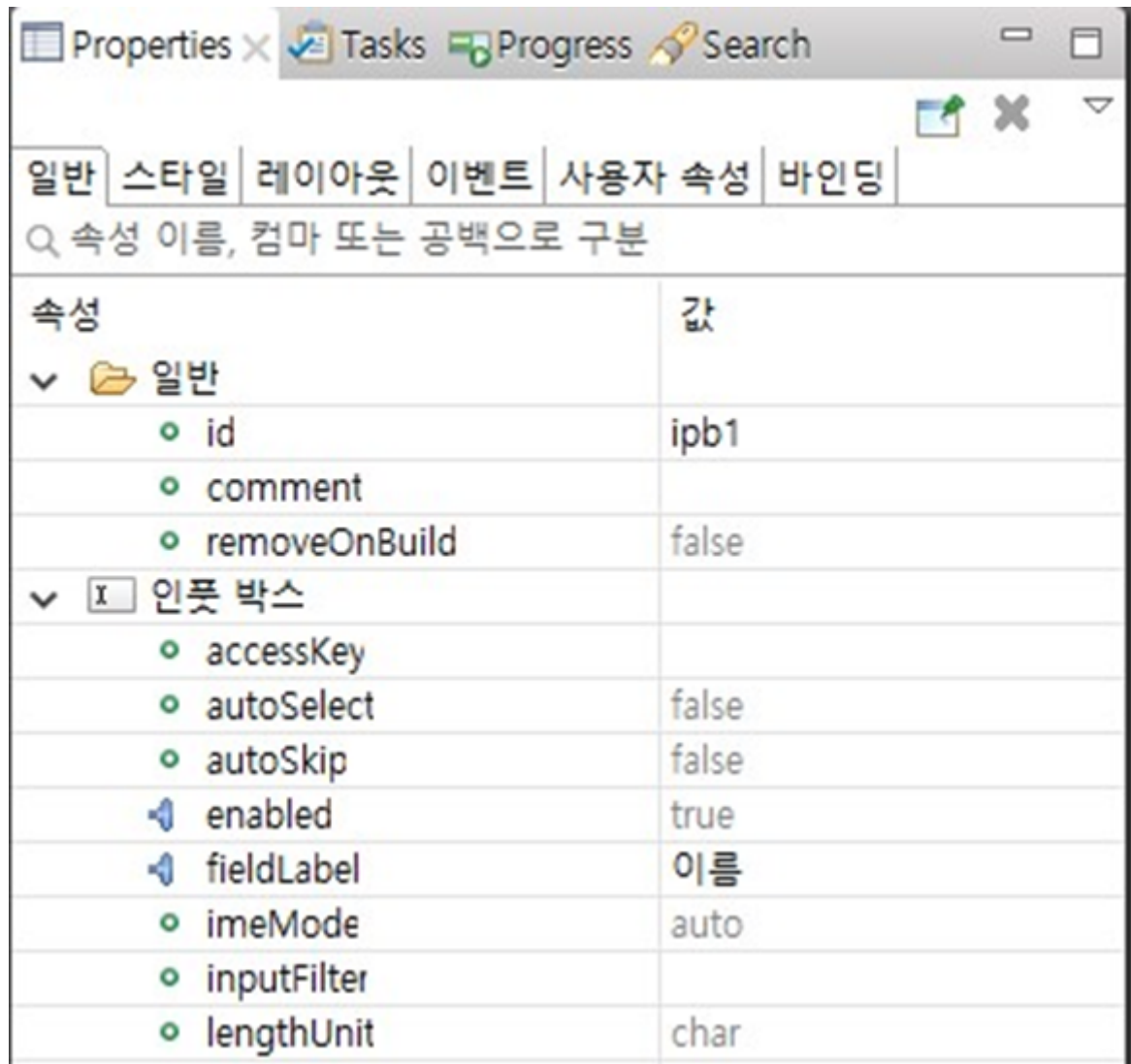
3.1.4. 인풋박스 / 넘버에디터 / 마스트에디터 / 텍스트에리어

- 인풋박스 컨트롤을 디자인 탭에 배치한 후 대체 텍스트를 설정한다

3.1.4.1. 대체 텍스트 처리

- 컨트롤의 fieldLabel 속성을 설정하여 필드 이름을 컨트롤에서 리더에서 읽을 수 있다

<그림 42. 인풋박스 fieldLabel 설정>



관련 항목: 2.1. 적절한 대체 텍스트 제공 - 1, 2, 4

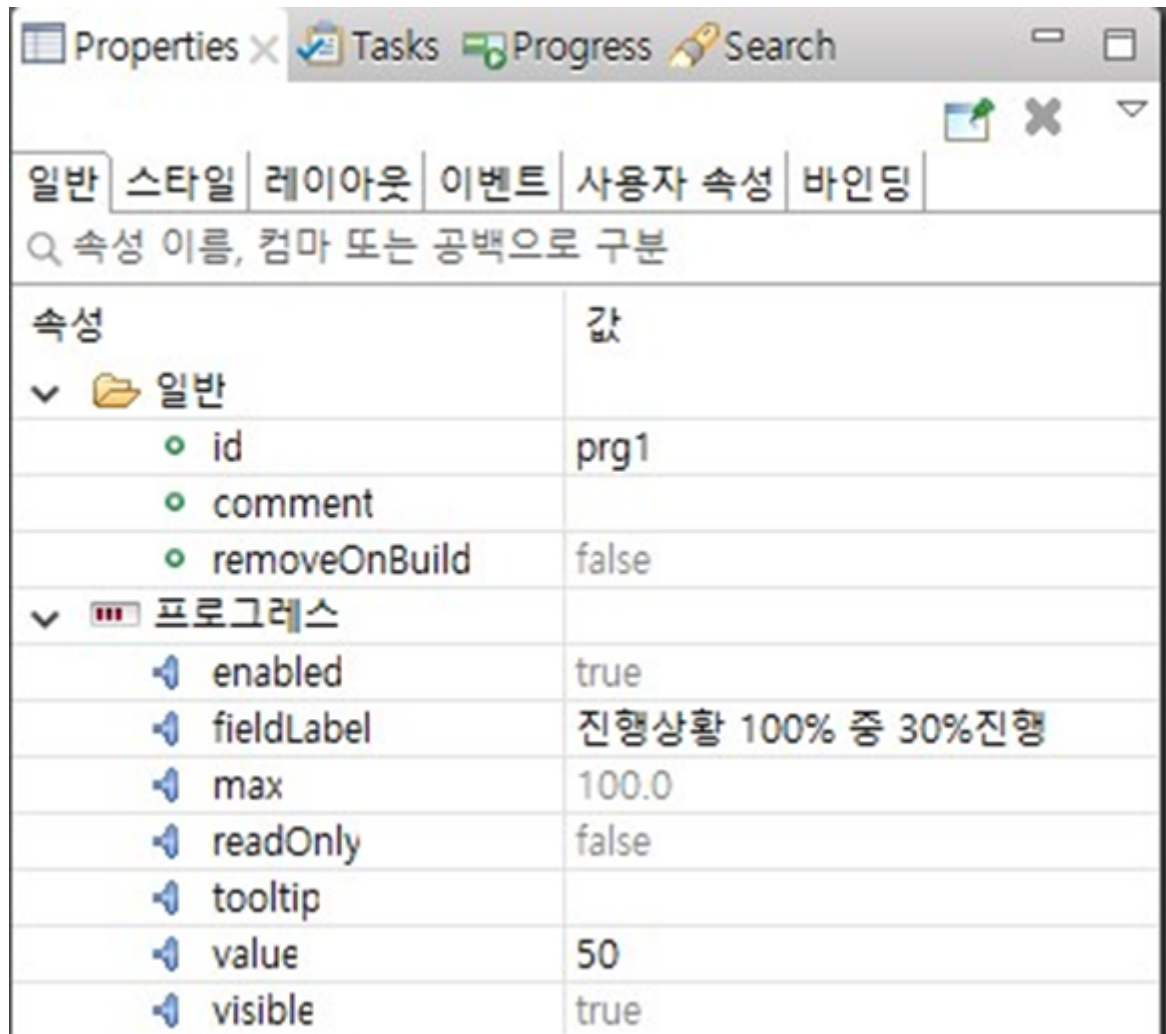
3.1.5. 프로그레스

- 프로그레스 컨트롤을 디자인 탭에 배치한 후 대체 텍스트를 설정한다

3.1.5.1. 대체 텍스트 처리

- fieldLabel 속성에서 진행상황을 알려주는 텍스트를 설정한다

<그림 43. 프로그레스 fieldLabel 설정>

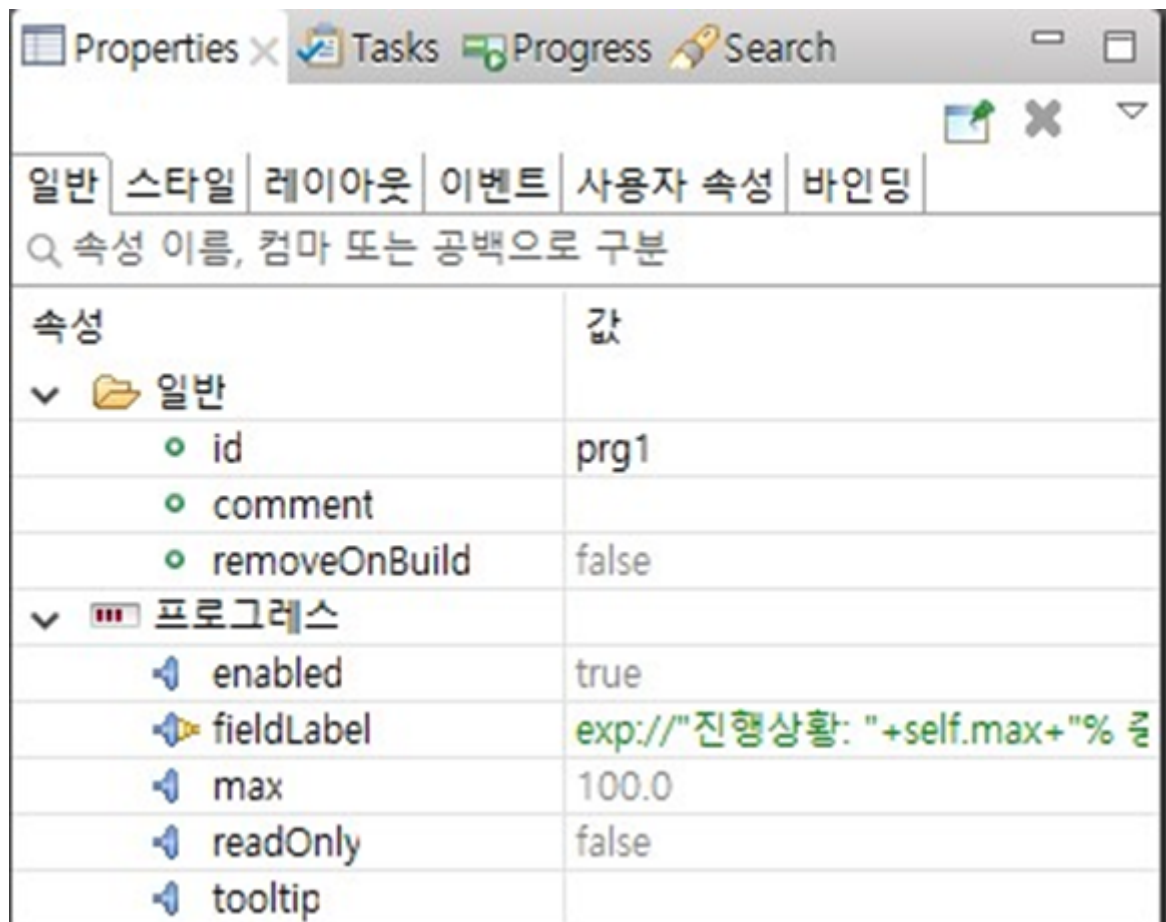


관련 항목: 2.1. 적절한 대체 텍스트 제공 - 1, 2, 4

3.1.5.2. 바인딩

- 진행되는 수치의 값을 익스프레션을 통해 동적으로 표시한다

<그림 44. 프로그레스 fieldLabel 익스프레션 바인딩>



<소스 29. 익스프레션 바인딩 설정 값>

```
"진행상황: "+self.max+"% 중 현재 "+value+"% 진행 중"
```

관련 항목 : 2.1. 적절한 대체 텍스트 제공 - 3

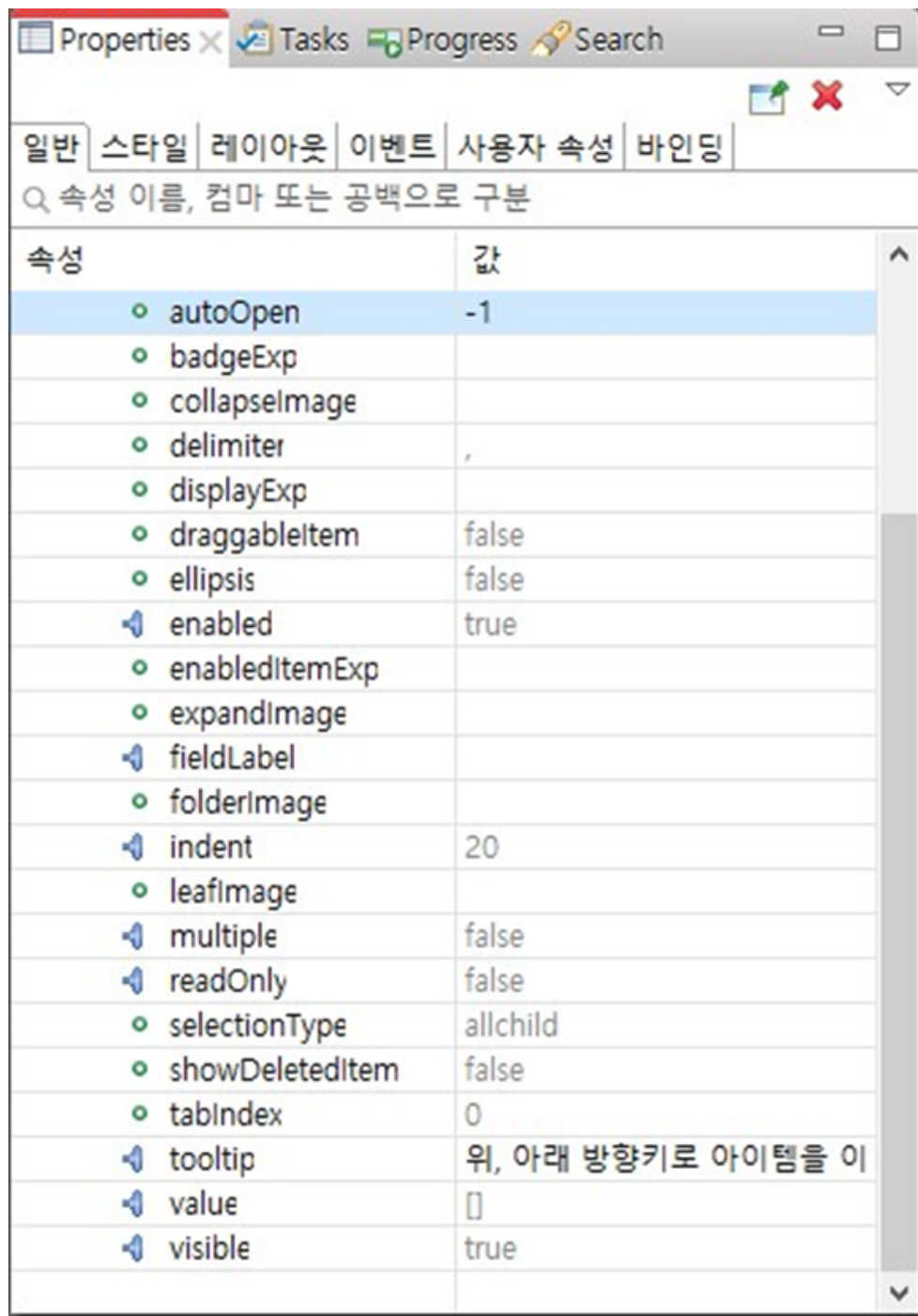
3.1.6. 트리

- 트리 컨트롤을 디자인 탭에 배치한 후 대체 텍스트를 설정한다

3.1.6.1. 대체 텍스트 처리

- 트리 컨트롤의 tooltip 속성을 통해 키보드 동작을 설명합니다
- fieldLabel 속성을 통해 필드의 이름을 리더에서 읽을 수 있다

<그림 45. 트리 tooltip 설정>



관련 항목: 2.1. 적절한 대체 텍스트 제공 - 1, 2, 4

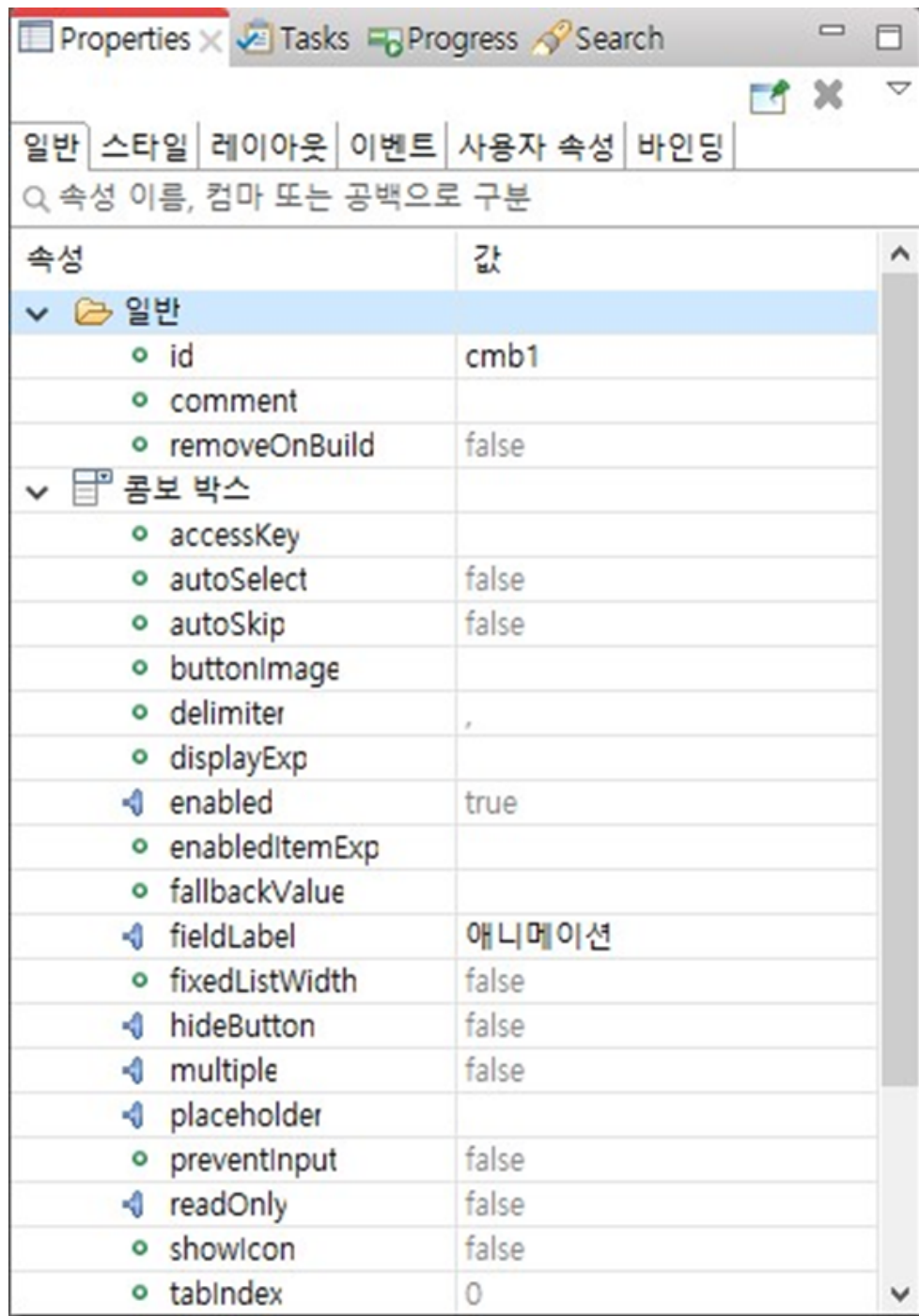
3.1.7. 콤보박스 / 링크드콤보박스 / 리스트박스

- 콤보박스 컨트롤을 디자인 탭에 배치한 후 대체 텍스트를 설정한다

3.1.7.1. 대체 텍스트 처리

- 컨트롤의 필드이름을 스크린리더에 읽을 수 있게 한다

<그림 46. 콤보박스 fieldLabel 설정>



관련 항목: 2.1. 적절한 대체 텍스트 제공 - 1, 2, 4

3.2. 키보드 사용 보장

3.2.1. 초점 제어

- eXbuilder6에서는 각 컨트롤의 초점제어를 세 가지 방식으로 지원한다

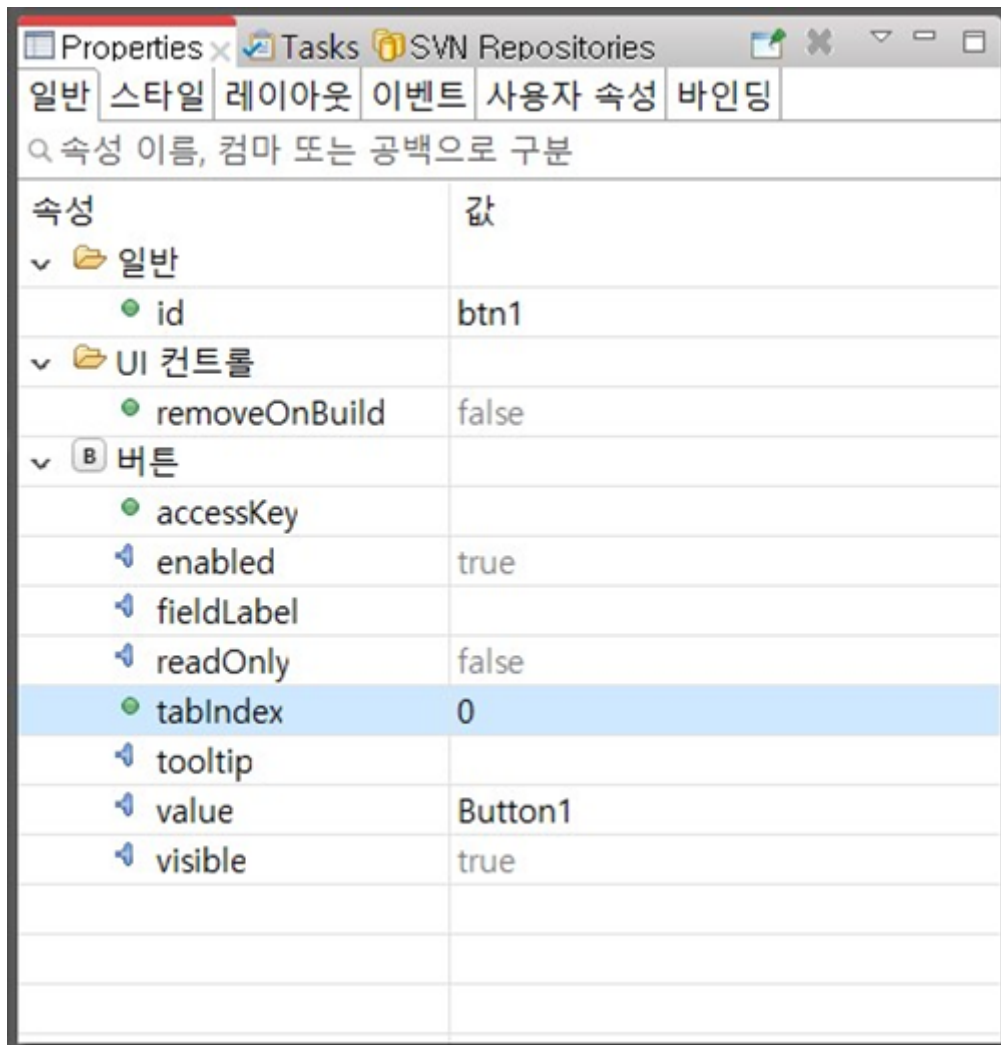
1) tabIndex를 통한 초점 제어

- Properties 탭의 tabIndex 속성을 0 이외의 숫자로 설정하여 물리적인 초점 순서를 제어할 수 있다
- TabIndex 값으로 초점을 제어하기 위해서는 초점을 받을 수 있는 모든 컨트롤이 논리적 모순이 없는 0 이상의 값으로 초점 순서가 정의되어야 하기 때문에 UDC 등을 사용하는 경우 초점 순서에 문제가 일어날 수 있으므로 수정하는 것을 권장하지 않는다

2) 컨트롤의 순서에 의한 초점 제어

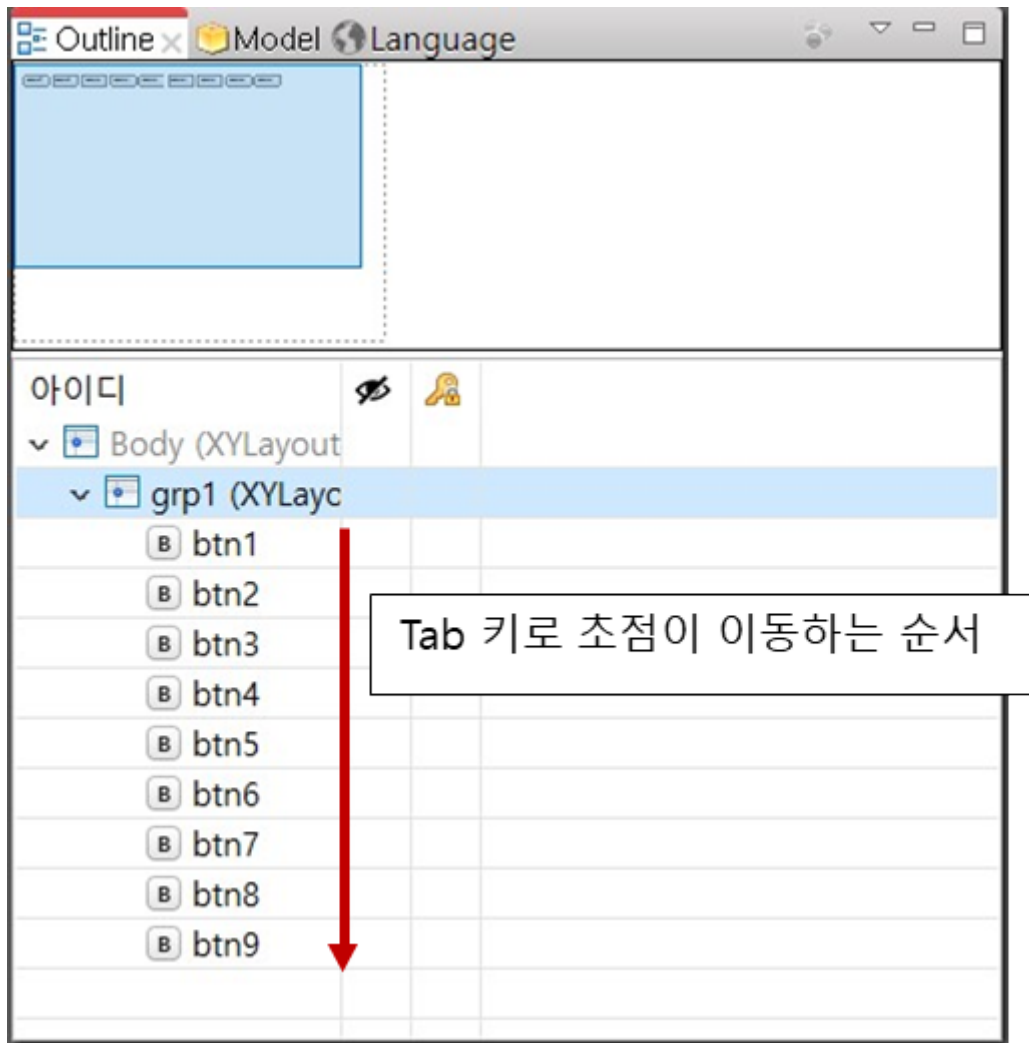
- 기본적으로 초점은 Outline View의 컨트롤 순서대로 이동된다
- 컨트롤의 순서에 의한 제어는 UDC를 포함한 App에서도 논리적인 문제의 소지가 최소화될 수 있기 때문에 권장된다
- 컨트롤의 순서는 Outline View를 통해 조정이 가능하다

<그림 47. 컨트롤의 tabIndex 설정>



- 컨트롤의 tabIndex 속성은 기존의 컨트롤 순서대로 초점이 맞춰질 수 있도록 0으로 설정한다(기본적으로 포커스를 받을 수 있는 컨트롤은 0으로 설정되어 있다)

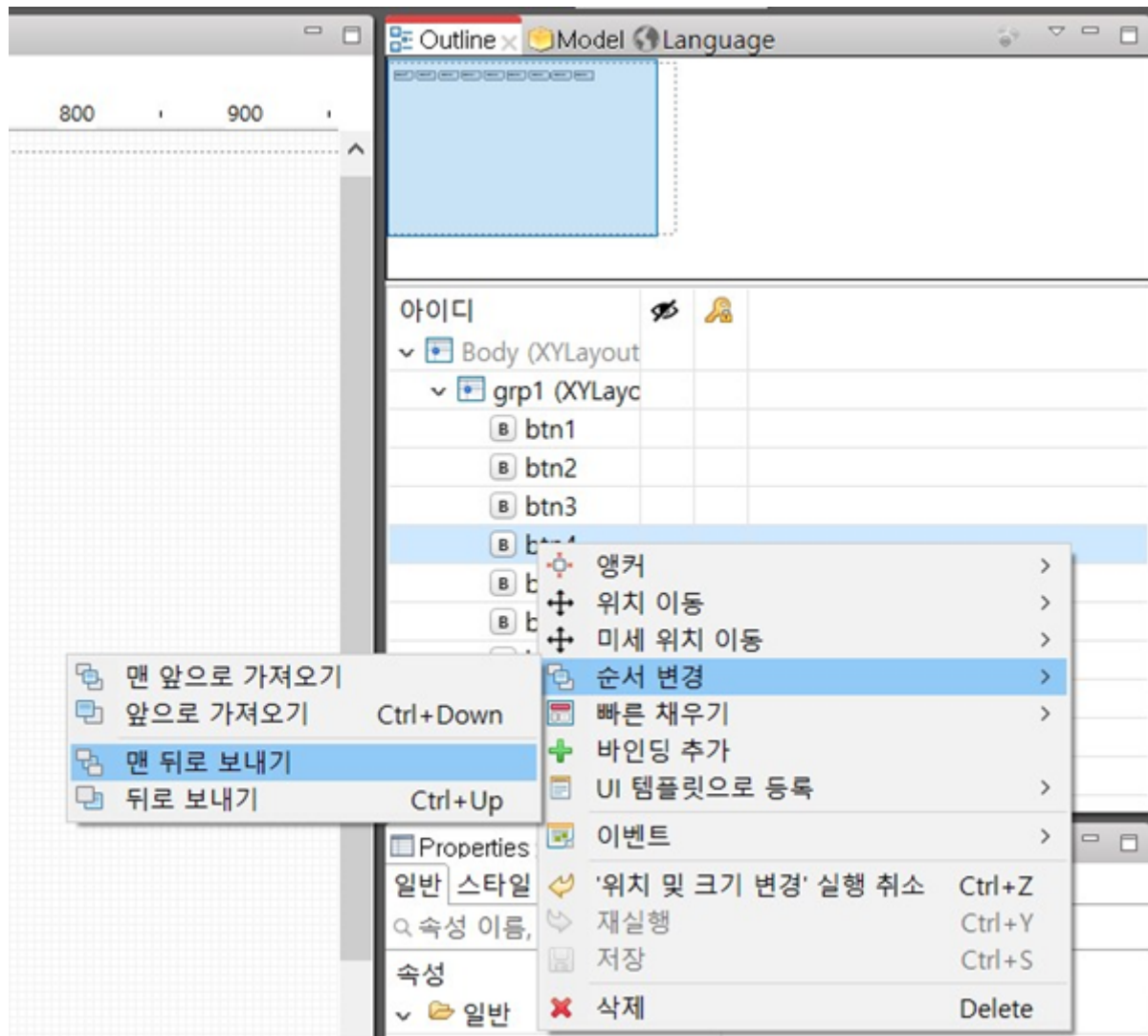
<그림 48. Outline View에 순서대로 정렬된 컨트롤>



• 포커스 되는 순서를 변경하는 방법

1. Outline View의 해당 컨트롤을 우 클릭하여 순서 변경을 할 수 있다.
2. 컨트롤을 선택한 뒤 ctrl+↑ 또는 ctrl+↓ 로 순서를 변경할 수 있다.

<그림 49. 컨트롤 초점 순서 변경>

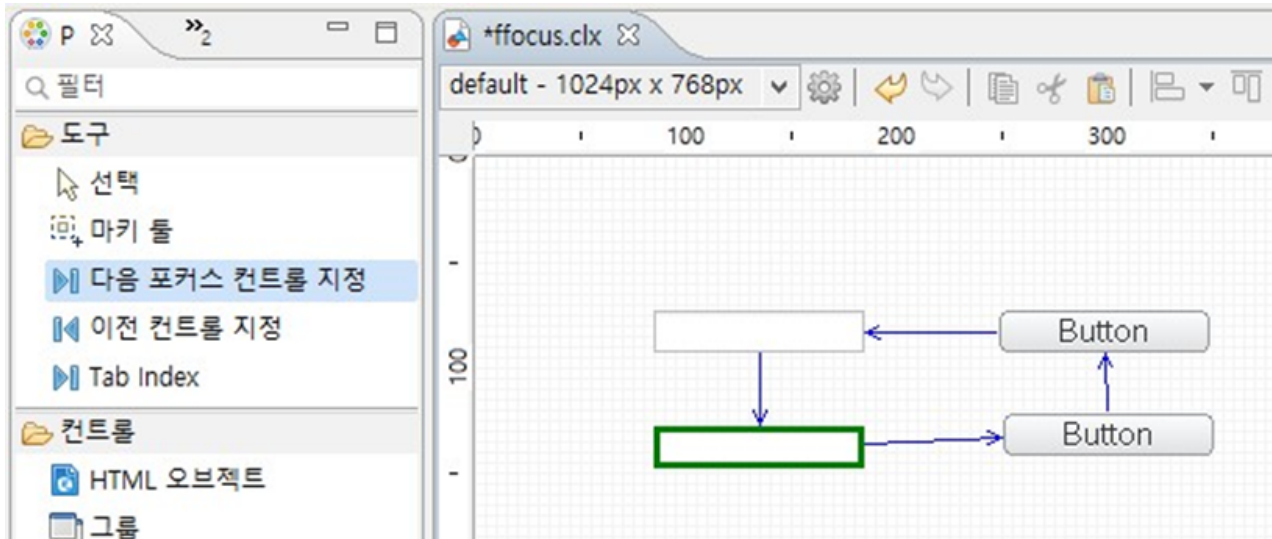


3) 다음/이전 포커스 컨트롤 지정

- 순서에 의한 제어가 어려운 특수한 경우 컨트롤 별로 이 컨트롤 다음에 초점을 받을 컨트롤을 강제로 지정할 수 있다

i 이 경우 초점이 순환이동 되지 않도록 주의해야 한다

<그림 50. 다음/이전 컨트롤 지정>



- 스튜디오의 Palette 창 도구 메뉴의 다음 포커스 컨트롤 지정을 선택한 상태로 순방향 초점 이동 순서를 지정할 수 있다
- 스튜디오의 Palette 창 도구 메뉴의 다음 포커스 컨트롤 지정을 선택한 상태로 역방향 초점 이동 순서를 지정할 수 있다
- 다음/이전 컨트롤 지정을 통해 순환구조의 초점 이동을 강제로 설정할 수 있고 순환하지 않는 경우 마지막 초점을 받은 컨트롤은 컨트롤의 순서 또는 tabIndex 순서에 의해 다음 컨트롤로 초점이 이동된다

관련 항목: 2.8. 키보드 사용 보장 - 1, 2 | 2.9. 초점 이동 - 2,6 | 2.19. 콘텐츠의 선형화 - 2

3.2.2. 다이얼로그

- 버튼을 클릭하여 다이얼로그 호출하기 위해 click 이벤트를 설정한다
- 다이얼로그 열기 팝업 시 초점은 팝업 최상단(그룹컨트롤)에 자동 포커스 되어야 하며 팝업 닫기 버튼을 눌렀을 때 기존 열기 버튼으로 포커스를 회귀해야 한다

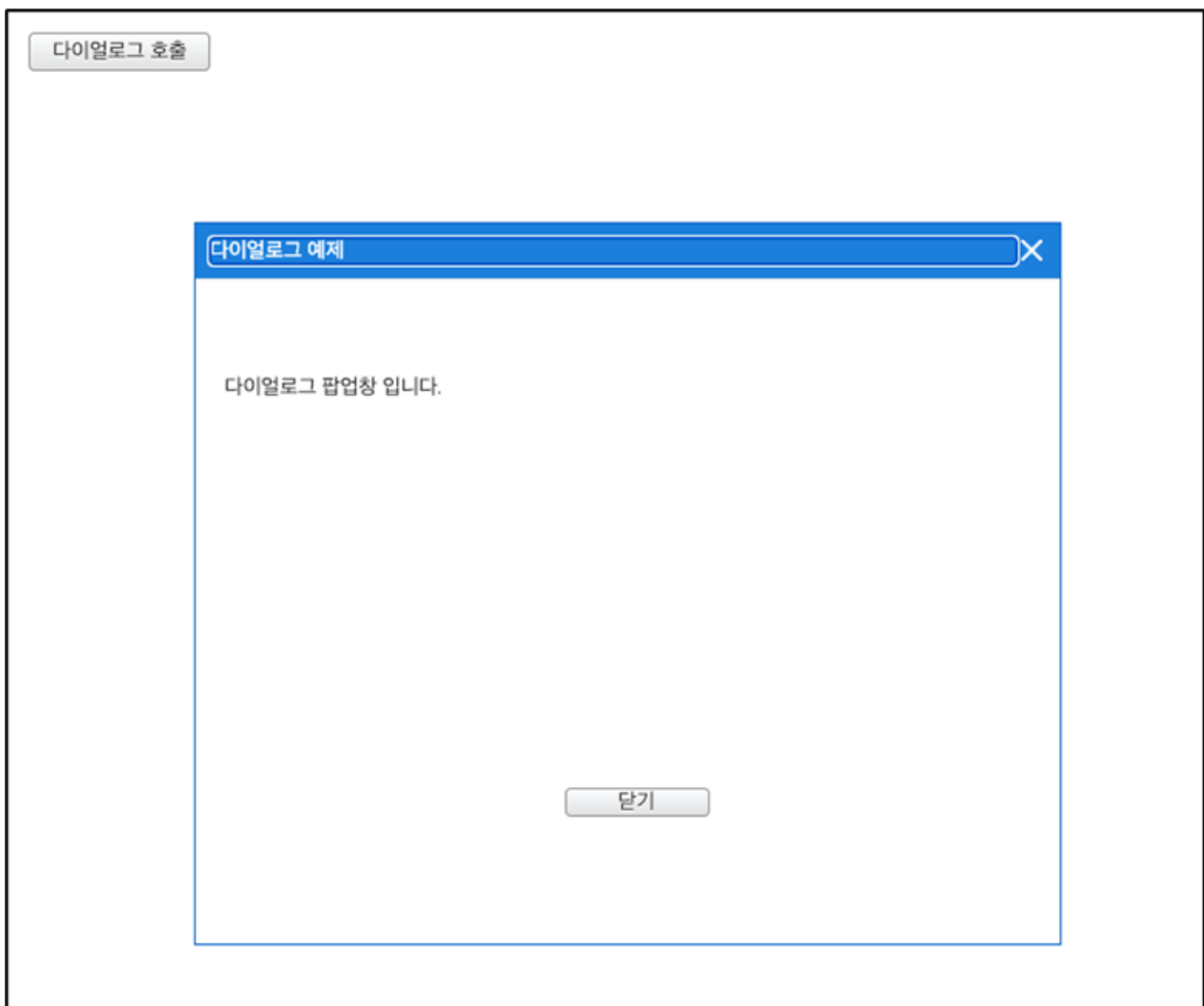
<소스 30. 다이얼로그 호출을 위한 click 이벤트 설정>

```
/*
 * "다이얼로그 호출" 버튼에서 click 이벤트 발생 시 호출.
 * 사용자가 컨트롤을 클릭할 때 발생하는 이벤트.
 */
function onBtn1Click( /* cpr.events.CMouseEvent */ e) {
    /**
     * @type cpr.controls.Button
     */
}
```

```
var btn1 = e.control;
app.openDialog("app/Dialog/popup", {width: 700, height: 500}, function(dialog) {
    dialog.ready(function(dialogApp) {
        // 다이얼로그에 포커스
        dialog.focus();
    });
    dialog.addEventListener("close", function(e){
        btn1.focus();
    });
}).then(function(returnValue) {

});
}
```

<그림 51. 다이얼로그 호출 및 포커스 설정>



<소스 31. '닫기'버튼의 클릭 이벤트 및 키보드 이벤트 설정>

```

/*
 * "닫기" 버튼에서 click 이벤트 발생 시 호출.
 * 사용자가 컨트롤을 클릭할 때 발생하는 이벤트.
 */
function onBtn1Click( /* cpr.events.CMouseEvent */ e) {
    /**
     * @type cpr.controls.Button
     */
    var btn1 = e.control;
    // 다이얼로그 닫기
    app.close();
}

/*
 * "닫기" 버튼에서 keydown 이벤트 발생 시 호출.
 * 사용자가 키를 누를 때 발생하는 이벤트.
 */
function onBtn1Keydown( /* cpr.events.CKeyboardEvent */ e) {
    /**
     * @type cpr.controls.Button
     */
    var btn1 = e.control;
    if (e.keyCode == cpr.events.KeyCode.ENTER) {
        onBtn1Click(e);
    }
}

```

- keydown 이벤트를 통해 엔터키를 누르면 클릭 이벤트를 호출할 수 있다

관련 항목: 2.9. 초점 이동 - 1 | 2.18. 사용자 요구에 따른 실행 - 2

3.2.3. 컨트롤의 내부 이동 단축키 많은 컨트롤이 내부 이동 단축키를 지원하고 있다.

(“Help > Help Contents > eXbuilder6 > 앱 개발 > 컨트롤 > UI 컨트롤”에서 각 컨트롤 별 키 동작에 대한 도움말을 확인 가능)

3.2.3.1. 콤보 박스 컨트롤

- 드롭다운 리스트 : Alt + ↓
- 아이템 이동 : 위, 아래 화살표
- 아이템 선택 : enter 키

3.2.3.2. 네비게이션바 컨트롤

- 메뉴 아이템 이동 : 위, 아래, 양 옆 화살표
- 아이템 선택 : enter 키

3.2.3.3. 트리 컨트롤

- 메뉴 아이템 이동 : 위, 아래, 양 옆 화살표
- 아이템 선택 : enter 키 또는 space 키

3.2.3.4. 슬라이더 컨트롤

- 슬라이드 증가 : 위, 오른쪽 화살표
- 슬라이더 감소 : 아래, 왼쪽 화살표

3.2.3.5. 탭 폴더, MDI 폴더 컨트롤

- TabItem 간 초점 이동 : 좌우 화살표
- 해당 TabItem 의 content 활성화 : enter 키 또는 아래 방향키
- 열려진 TabItem 의 content 로 초점 이동 : tab 키를 통해 이동
- content 내부에서는 tab 키를 통해 컨트롤 간 초점이동이 가능하다

3.2.3.6. 아코디언

- SectionItem 간 초점 이동 : 상하 화살표
- 해당 SectionItem 의 content 활성화 : enter 키
- 열려진 SectionItem 의 content 로 초점 이동 : tab 키를 통해 이동

- content 내부에서는 tab 키를 통해 컨트롤 간 초점 이동이 가능하다

3.2.3.7. 파일업로드

- tab 키로 체크박스, 버튼 순으로 이동이 가능하다

3.2.3.8. 그리드

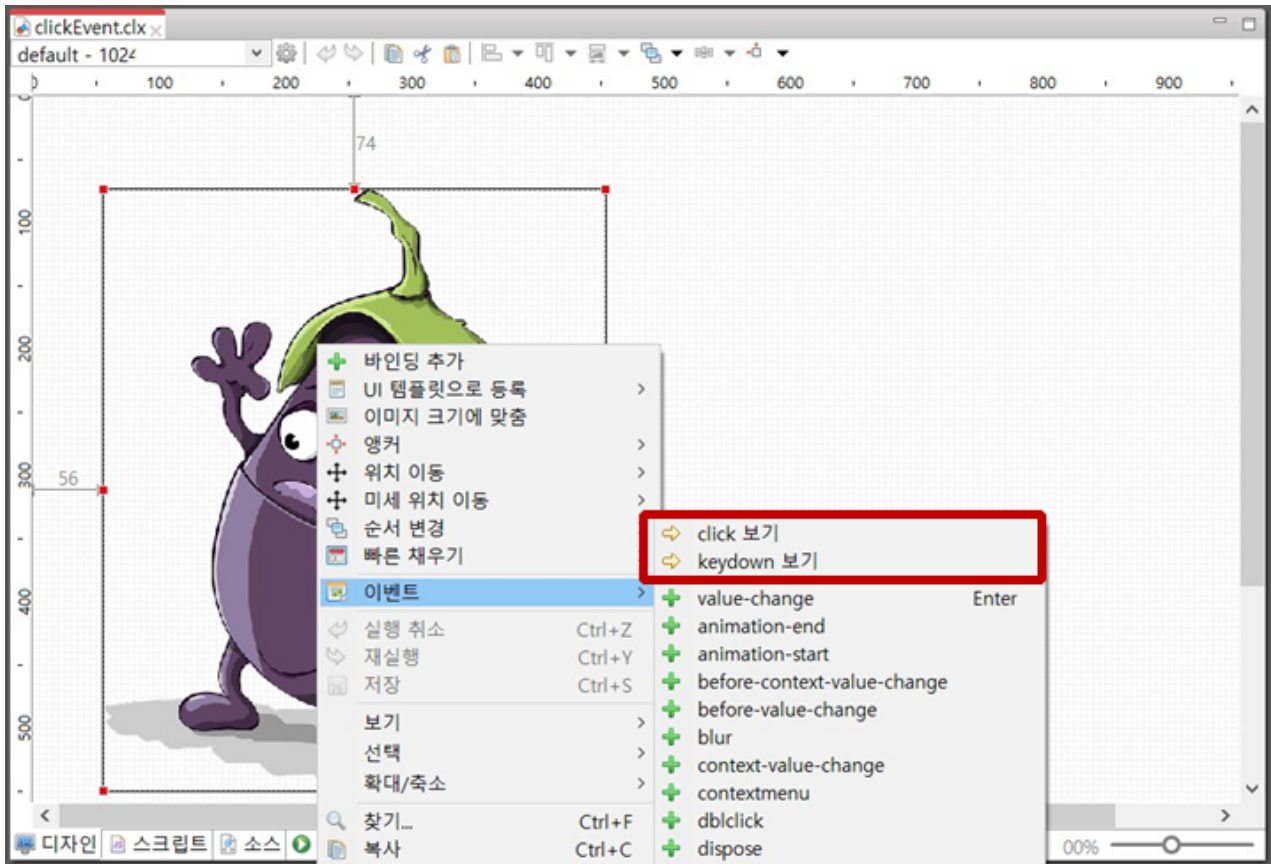
- 트리그리드의 확장 : Ctrl + →
- 트리그리드의 축소 : Ctrl + ←

관련 항목 : 2.8. 키보드 사용 보장 - 1 | 2.9. 초점 이동 - 1 | 2.18. 사용자 요구에 따른 실행 - 3

3.2.4. onclick 이벤트 및 onkeydown 이벤트

- onkeydown 이벤트가 실행되면 onclick 이벤트가 호출될 수 있도록 설정할 수 있다

<그림 52. 이미지 컨트롤의 이벤트 설정>



<소스 32. 클릭 또는 엔터키를 눌렀을 때 이벤트 설정>

```

/*
 * 이미지에서 click 이벤트 발생 시 호출.
 * 사용자가 컨트롤을 클릭할 때 발생하는 이벤트.
 */
function onImg1Click( /* cpr.events.CMouseEvent */ e) {
    /**
     * @type cpr.controls.Image
     */
    var img1 = e.control;
    alert("캐릭터 이미지입니다");
}
/*
 * 이미지에서 keydown 이벤트 발생 시 호출.
 * 사용자가 키를 누를 때 발생하는 이벤트.
 */
function onImg1Keydown( /* cpr.events.CKeyboardEvent */ e) {
    /**
     * @type cpr.controls.Image
     */
    var img1 = e.control;

```

```
if (e.keyCode == cpr.events.KeyCode.ENTER) {  
    onImg1Click(e);  
}  
}
```

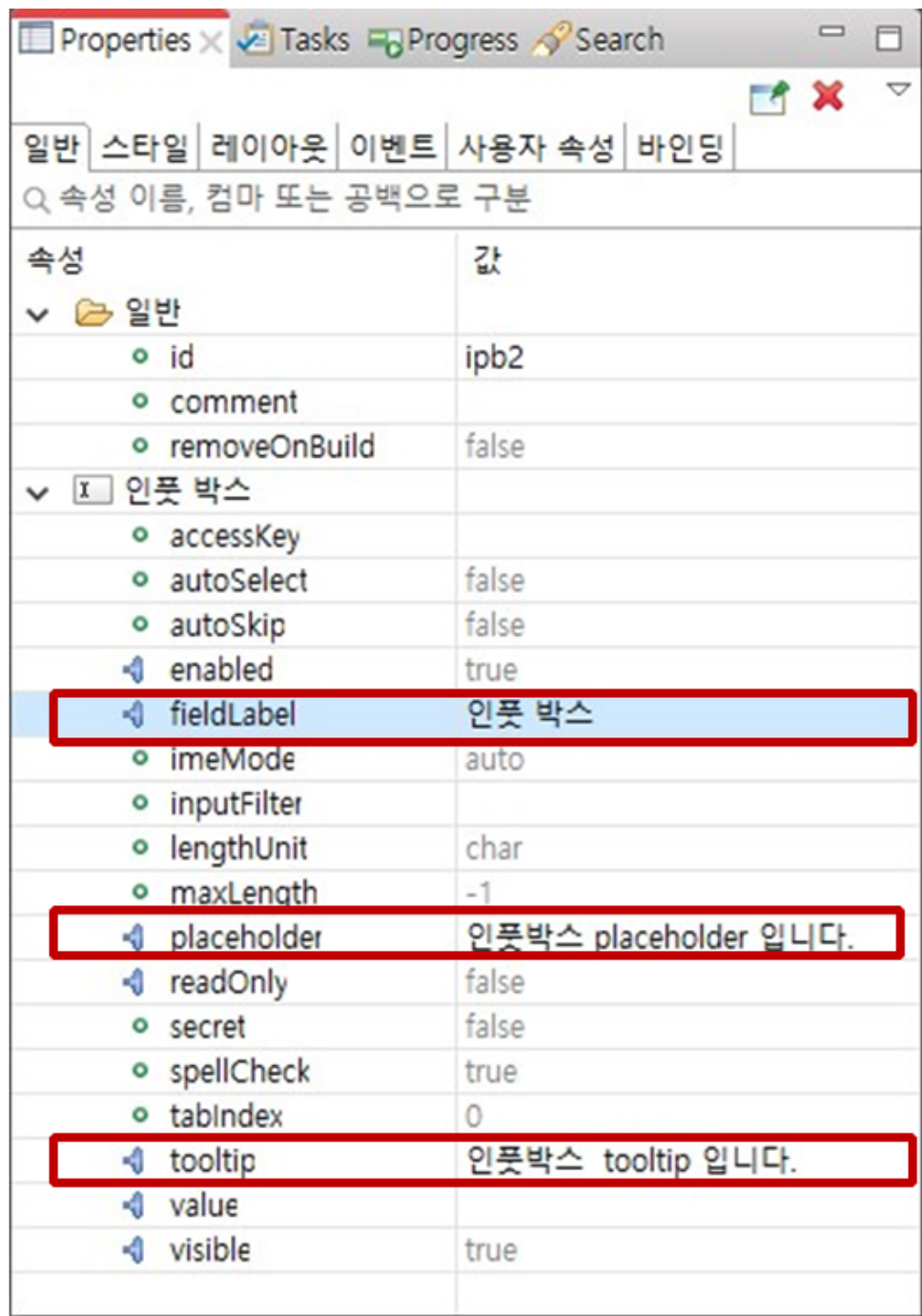
관련 항목: 2.8. 키보드 사용 보장 - 1 | 2.9. 초점 이동 - 3,4

3.3. 레이블/타이틀 제공

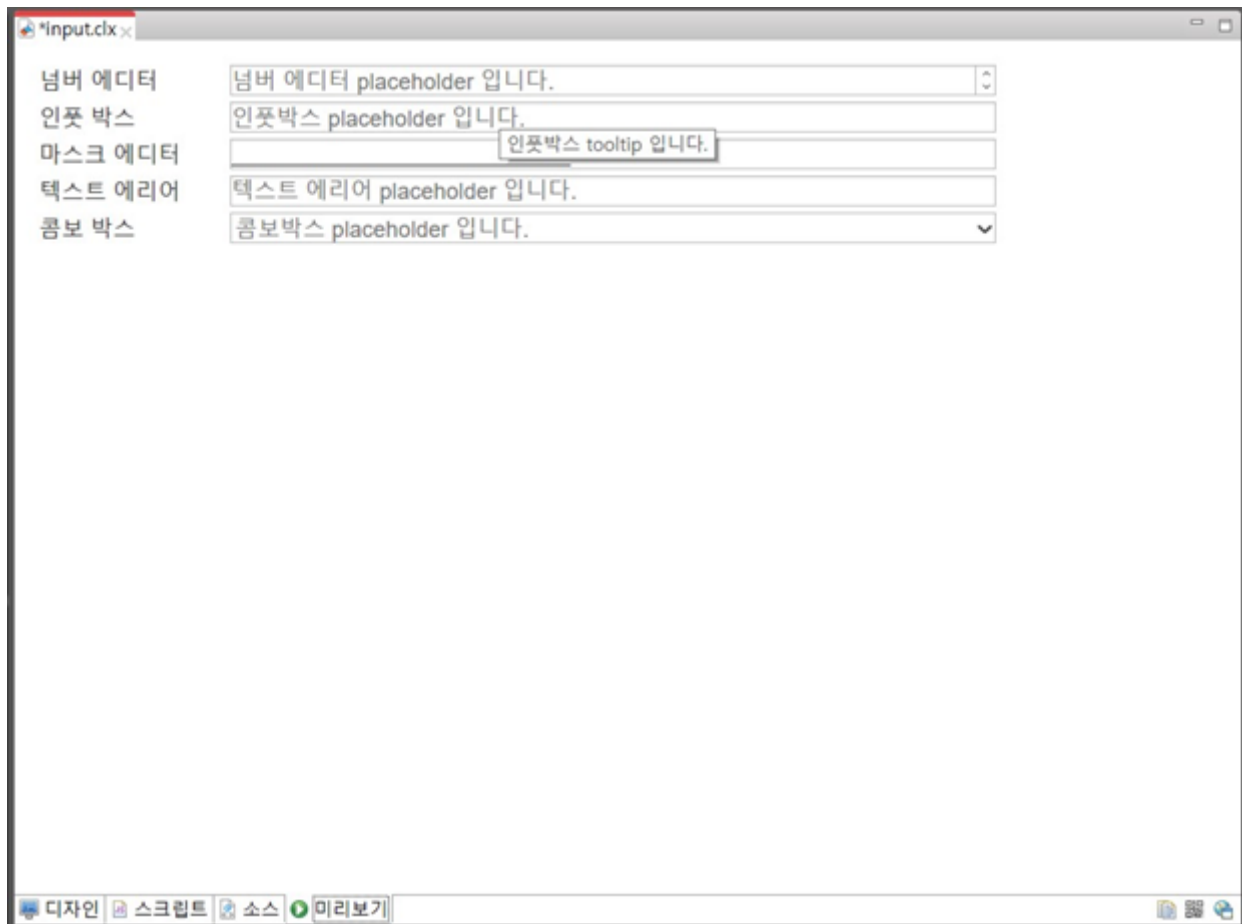
3.3.1. 입력 계열 컨트롤의 레이블 활용 가이드

- 입력 계열 컨트롤의 레이블 속성(tooltip, fieldLabel, placeholder)를 설정할 수 있다
- 마스크 에디터 컨트롤은 placeholder 속성을 지원하지 않는다
- tooltip 속성은 HTML Tag의 title 속성으로 표현된다
- placeholder 속성은 HTML Tag의 placeholder 속성으로 표현된다
- fieldLabel 속성은 HTML Tag의 aria-label 속성으로 표현된다
- 그리드 셀에 배치된 컨트롤의 fieldLabel은 헤더 셀의 text로 작성한다

<그림 53. 인풋 박스 컨트롤의 레이블 설정>



<그림 54. 입력 계열 컨트롤 레이블 활용 결과>

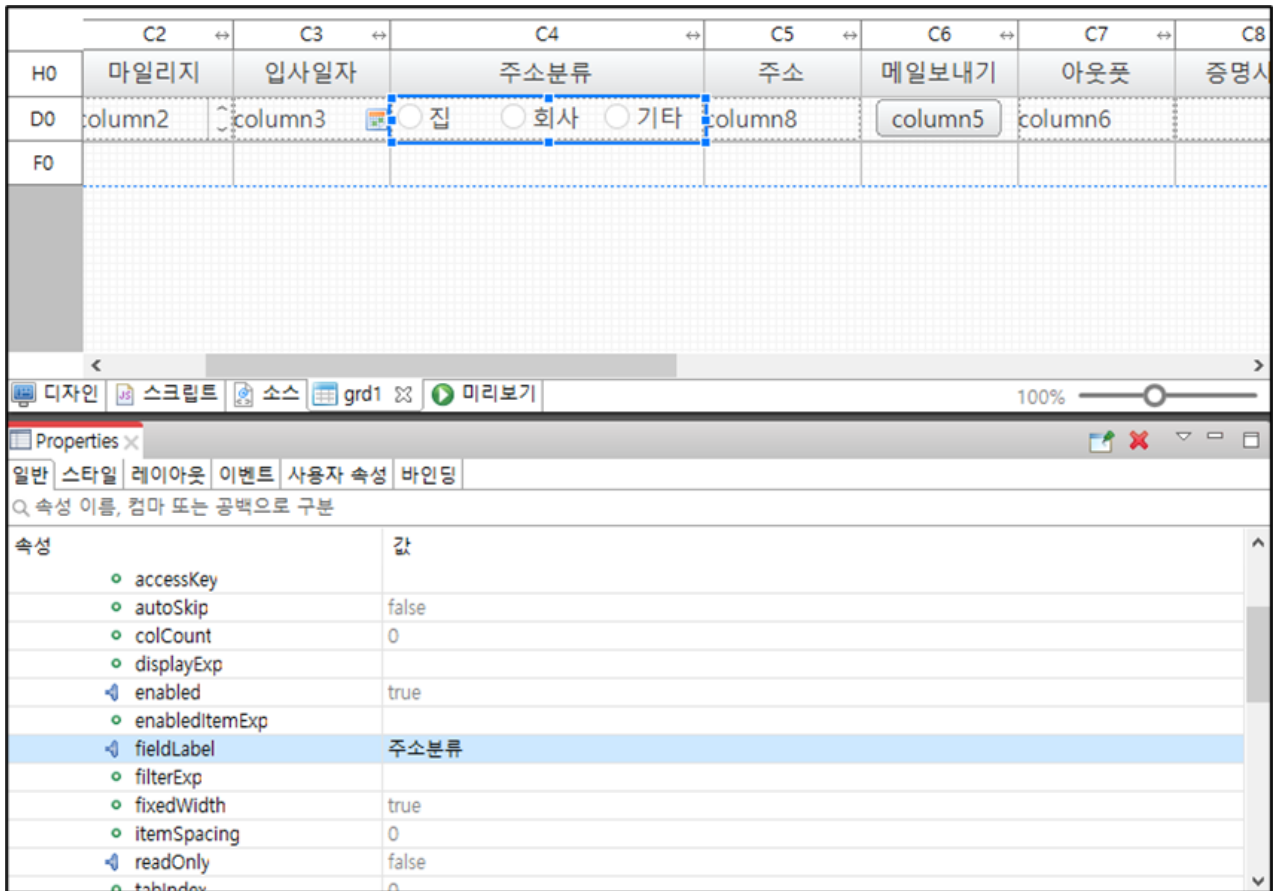


<소스 33. 입력 계열 컨트롤 tooltip 속성 지정 결과>

```
<input class="cl-text" style="ime-mode:auto"
  data-ndid="11c60c0f-0476-2fe4-b587-951a22686488" role="textbox"
  type="text" title="인풋박스 tooltip입니다." tabindex="0" arialabel="인풋 박스"
  placeholder="인풋박스 placeholder입니다.">
```

- tooltip으로 제공한 속성이 title 속성으로 확인할 수 있다

<그림 55. 그리드 셀에 배치된 컨트롤의 fieldLabel>



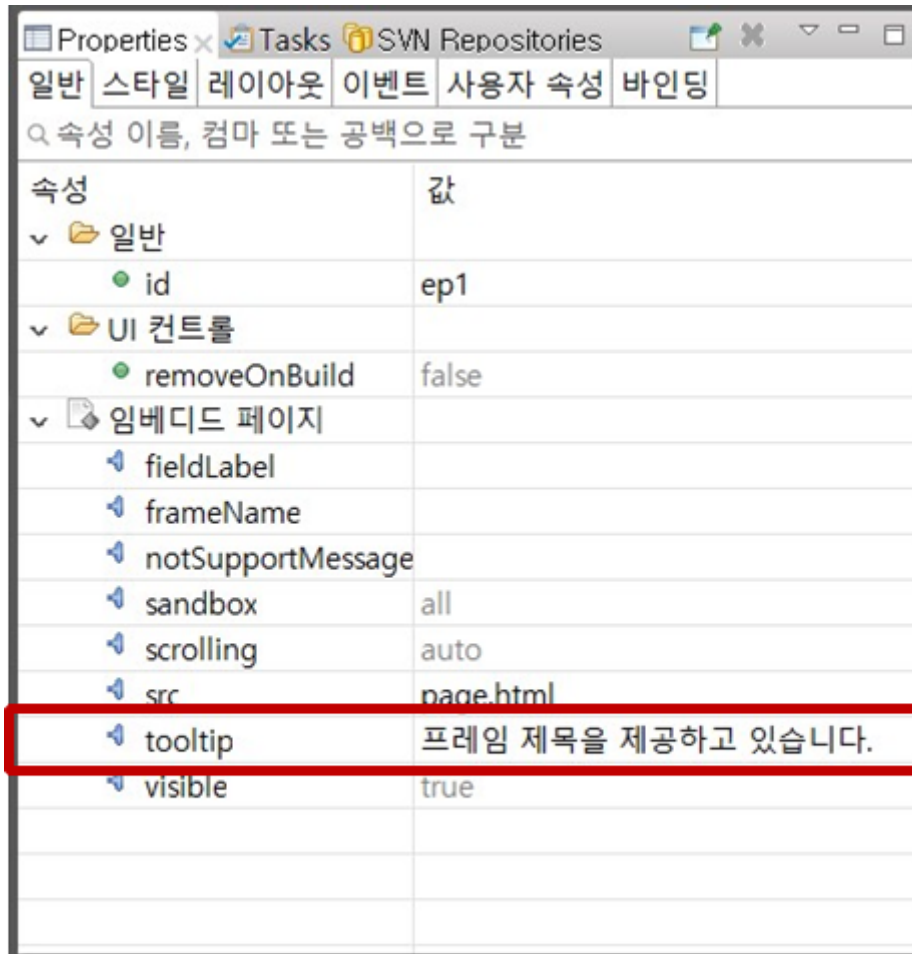
- 그리드에 배치된 컨트롤의 경우, fieldLabel은 해당 헤더 셀의 text로 작성한다

관련 항목: 2.4. 명확한 지시 사항 제공 - 1 | 2.21. 레이블 제공 - 2

3.3.2. 임베디드 페이지 컨트롤의 타이틀 입력

- 임베디드 페이지 컨트롤을 사용할 경우 반드시 tooltip 속성에 상황에 맞는 제목을 설정하여 임베디드 페이지가 표현되는 iframe 영역에 적절한 프레임 제목을 제공하여야 한다

<그림 56. 임베디드 페이지 컨트롤의 tooltip 속성 설정>



<소스 34. tooltip 속성 적용 결과>

```
<iframe style="position: absolute; height: 100%; width:100%; border:none"
  data-ndid="9717491f-9843-35b6-9e94-4de2a2b56d29" frameborder="0" scrolling="auto"
  src="app/EmbeddedPage/page.html" title="프레임 제목을 제공하고 있습니다."><p></p></iframe>
```

관련 항목: 2.15. 제목 제공 - 3

3.4. 반복영역 건너뛰기

3.4.1. 네비게이션바 컨트롤을 이용한 반복영역 건너뛰기

- 네비게이션 바에 포커스가 맞춰질 때, 엔터키를 누르면 바로 본문으로 이동할 수 있도록 설정할 수 있다.

<소스 35. 네비게이션바에서 keydown 이벤트 설정>

```
/*
 * 네비게이션 바에서 keydown 이벤트 발생 시 호출.
 * 사용자가 키를 누를 때 발생하는 이벤트.
 */
function onNgb1Keydown( /* cpr.events.CKeyboardEvent */ e) {
    /**
     * @type cpr.controls.NavigationBar
     */
    var ngb1 = e.control;
    if (e.keyCode == cpr.events.KeyCode.ENTER) {
        app.lookup("txa1").focus();
    }
}
```

<그림 57. Keydown 이벤트 적용 결과>



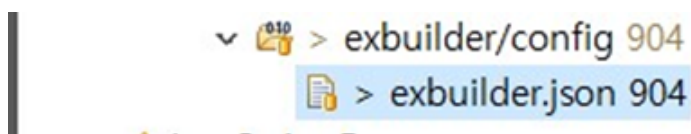
관련 항목: 2.8. 키보드 사용 보장 - 2 | 2.14. 반복 영역 건너뛰기 - 1

3.5. 기본 언어 표시

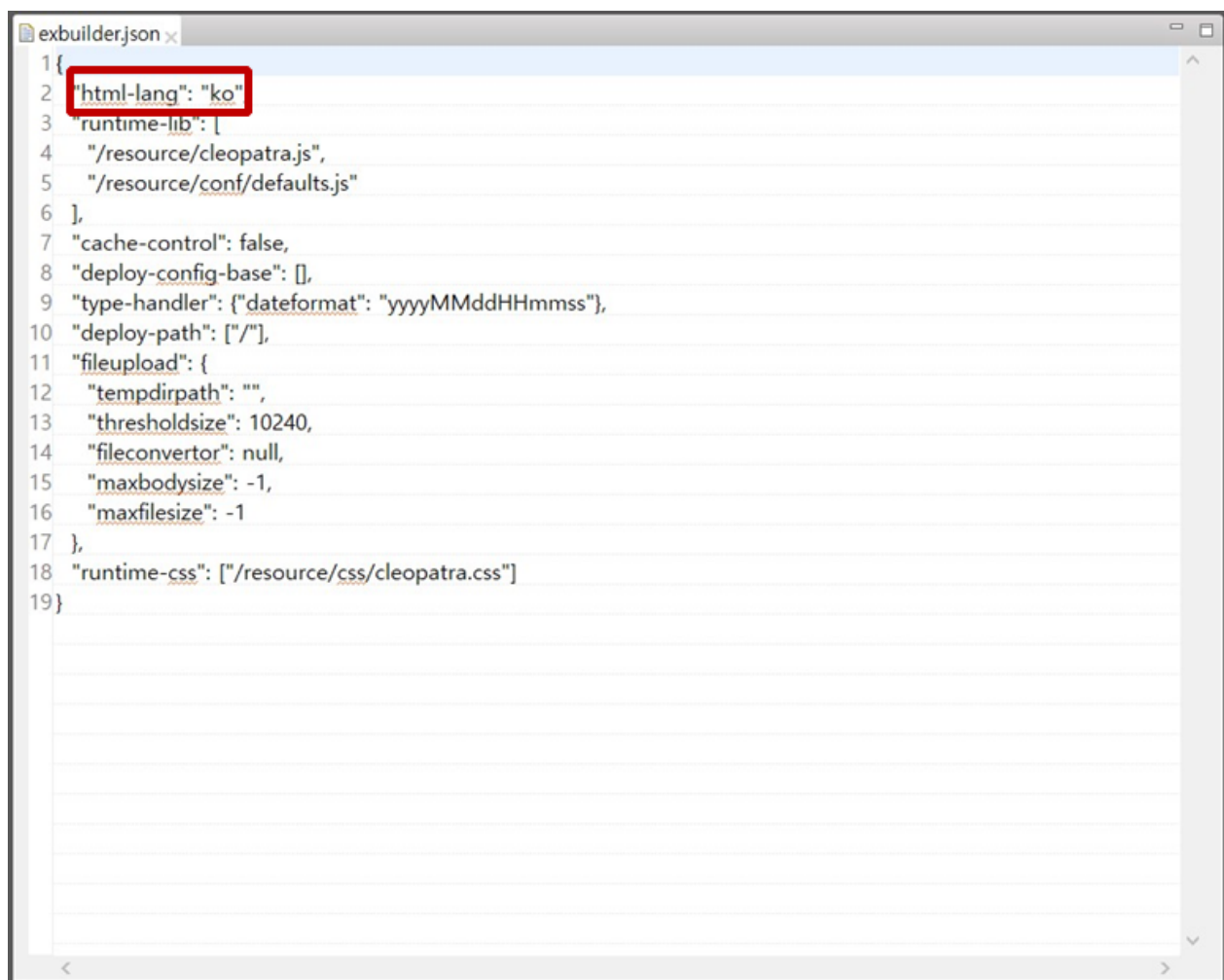
3.5.1. exbuilder.json 의 html-lang 속성을 이용한 기본 언어 설정

- exbuilder.json 파일은 DynamicWebProject에서 exbuilder 관련 환경설정 정보를 담고 있는 파일이다
- 경로 : /exbuilder/config/ exbuilder.json

<그림 58. exbuilder.json 파일 위치>



<그림 59. exbuilder.json 파일에서 html-lang 지정>



- 설정하고자 하는 나라의 언어 코드를 해당 부분에 입력하여 기본 언어를 지정할 수 있다

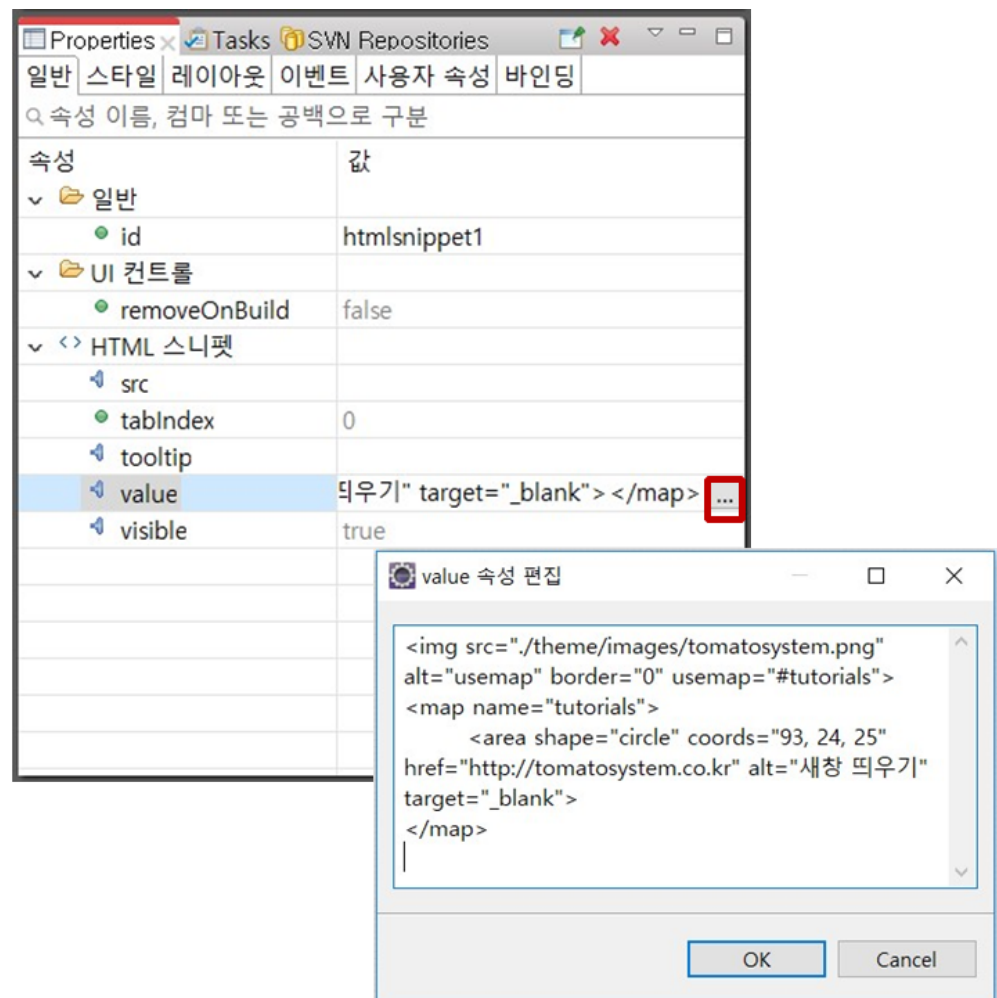
관련 항목: 2.17. 기본 언어 표시 - 1

3.6. 기타 마크업 삽입

3.6.1. HTML 스니펫 컨트롤의 활용

3.6.1.1. 이미지맵 처리

<그림 60. HTML 스니펫 컨트롤의 value 속성 설정>



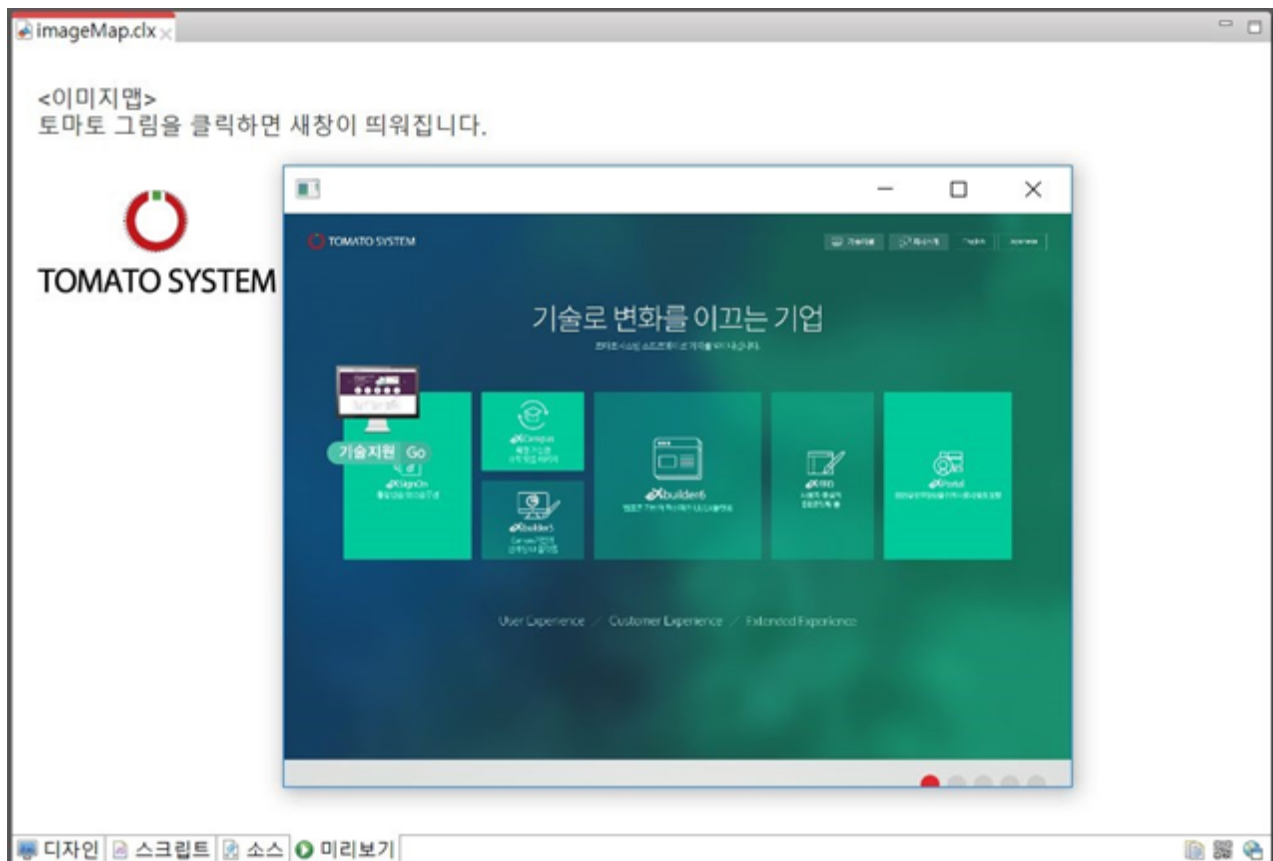
<소스 36. 이미지맵 value 속성 소스>

```

<map name="tutorials">
```

```
<area shape="circle" coords="93, 24, 25" href="http://tomatosystem.co.kr" alt="새창 띄우기"
target="_blank">
</map>
```

<그림 61. 이미지맵 설정 후 결과>



3.6.1.2. <a> 태그 처리

- HTML 스프레드 컨트롤의 value 속성에 a 태그를 작성하여 alt를 설정할 수 있다

<소스 37. img 태그 안에 직접 alt 설정>

```
<a></a>
```

<소스 38. alt 를 빈 값으로 제공했을 때 대체 텍스트 설정>

```
<a><span>자연풍경입니다!</span></a>
```

- HTML 스프레드 컨트롤의 value 속성에 a 태그를 활용하여 다운로드를 받을 수 있다

<소스 39. a 태그를 이용하여 멀티미디어 콘텐츠 다운로드>

```
<a href="./data/testVideo.mp4">동영상 다운로드</a>
```

<그림 62. 멀티미디어 콘텐츠 다운로드 결과>



동영상 다운로드

관련 항목 : 2.1. 적절한 대체 텍스트 제공 - 2, 4 | 2.2. 자막 제공 - 1, 3

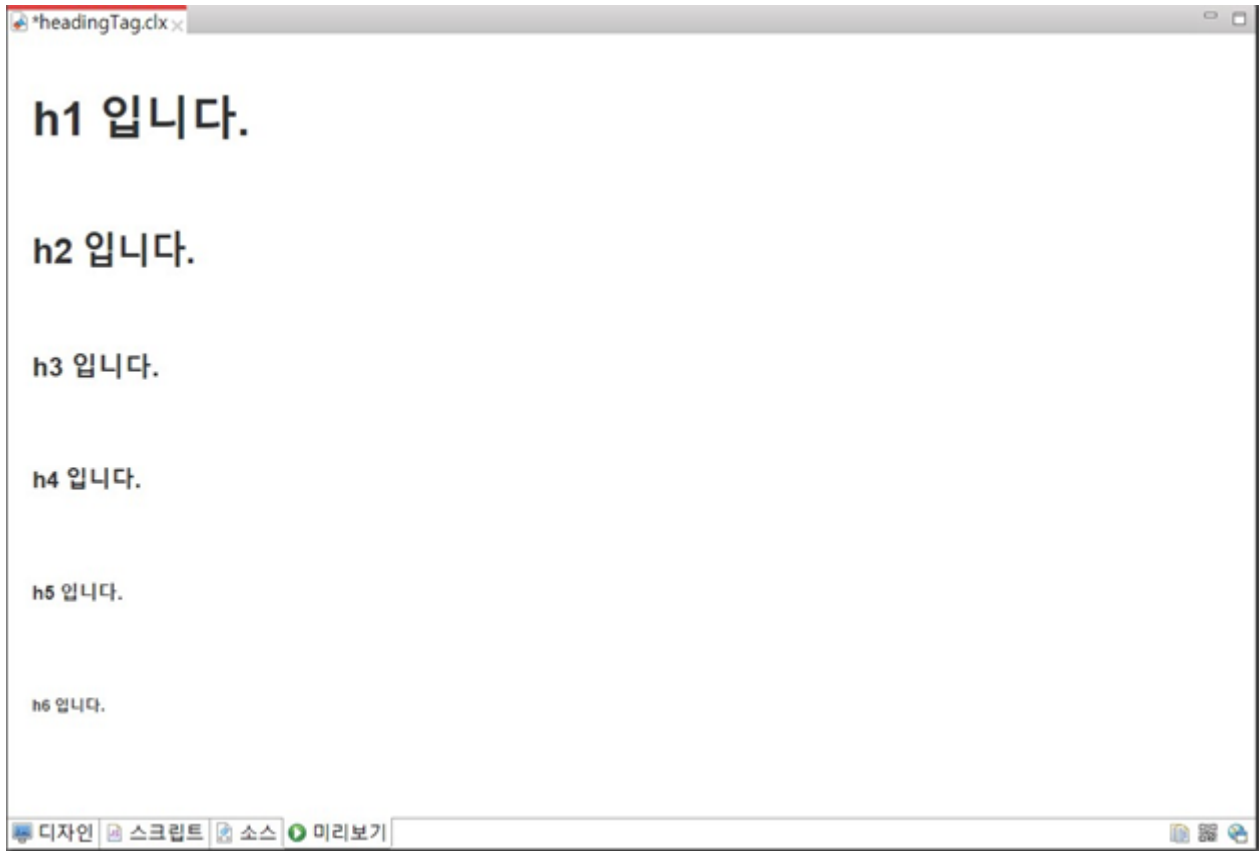
3.6.1.3. Heading Tag 의 활용

- HTML 스니펫 컨트롤의 value 속성에 Heading Tag 를 사용하여 결과를 확인할 수 있다

<소스 40. HTML 스니펫 컨트롤에 Heading Tag 설정>

```
<h1>h1 입니다.</h1><br />  
<h2>h2 입니다.</h2><br />  
<h3>h3 입니다.</h3><br />  
<h4>h4 입니다.</h4><br />  
<h5>h5 입니다.</h5><br />  
<h6>h6 입니다.</h6><br />
```

<그림 63. Heading Tag 결과>



관련 항목 : 2.15. 제목 제공 - 6