

전자정부 모바일 표준프레임워크 UX Component 실습

eGovFrame



Contents

1. _ Start.
2. _ Step 1. Button & Icon
3. _ Step 2. Link & Dialog
4. _ Step 3. Tab, Panel & Panel Widget
5. _ Step 4. Form, Grid & Popup Widget

Start.

[Start 01] 'viewport' setting

[Start 02] Library Import

Start.

파일 경로 : C:/eGovFrame-3.10.0/workspace.edu/lab402-ux-component/src/main/webapp/WEB-INF/jsp/practice/step1.jsp

[Start 01] 'viewport' 설정

- 모바일 브라우저가 각각 다르기 때문에 이들 브라우저의 크기 별로 적절하게 표현하기 위해서 **viewport <meta> Tag** 를 <head> 영역에 삽입한다.
- **<meta> Tag 'content' 속성의 'width' 값을 지정하여 viewport 의 가로 크기를 디바이스 가로 크기로 설정한다.**
- **<meta> Tag 'content' 속성의 'initial-scale' 값을 지정하여 viewport 의 기본 확대비율을 100% 로 설정한다.**

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no" />
```

[참고] viewport meta tag 상세설명

속성명	설명	속성명	설명
width	viewport의 가로길이지정.	height	viewport의 픽셀 세로길이 지정
minimum-scale	사용자가 축소할 수 있는 최소 배율 지정	maximum-scale	사용자가 확대할 수 있는 최대 확대 비율
initial-scale	처음 모바일 페이지가 보여질 때, 초기 페이지 배율 지정	user-scalable	사용자의 배율을 확대 또는 축소 허용 여부 설정

Start.

[Start 02] Library Import

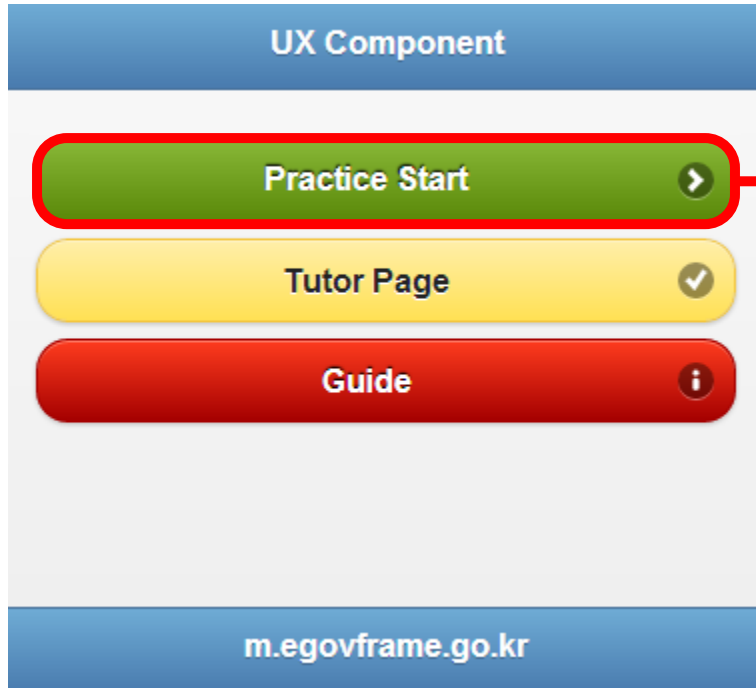
- ❑ 전자정부 모바일 표준프레임워크 실행 환경을 이용하기 위해 라이브러리를 적용한다.
- ❑ <link> Tag 를 이용하여 CSS 파일을 Import 한다.
- ❑ <script> Tag 를 이용하여 JavaScript 파일을 Import 한다.

```
<!-- eGovFrame Common import -->
<link rel="stylesheet" href="<c:url value='/css/egovframework/mbi/cmm/jquery.mobile-1.4.5.css' />" />
<link rel="stylesheet" href="<c:url value='/css/egovframework/mbi/cmm/EgovMobile-1.4.5.css' />" />
<script type="text/javascript" src="<c:url value='/js/egovframework/mbi/cmm/jquery-1.11.2.min.js' />"></script>
<script type="text/javascript" src="<c:url value='/js/egovframework/mbi/cmm/jquery.mobile-1.4.5.min.js' />"></script>
<script type="text/javascript" src="<c:url value='/js/egovframework/mbi/cmm/EgovMobile-1.4.5.js' />"></script>
```

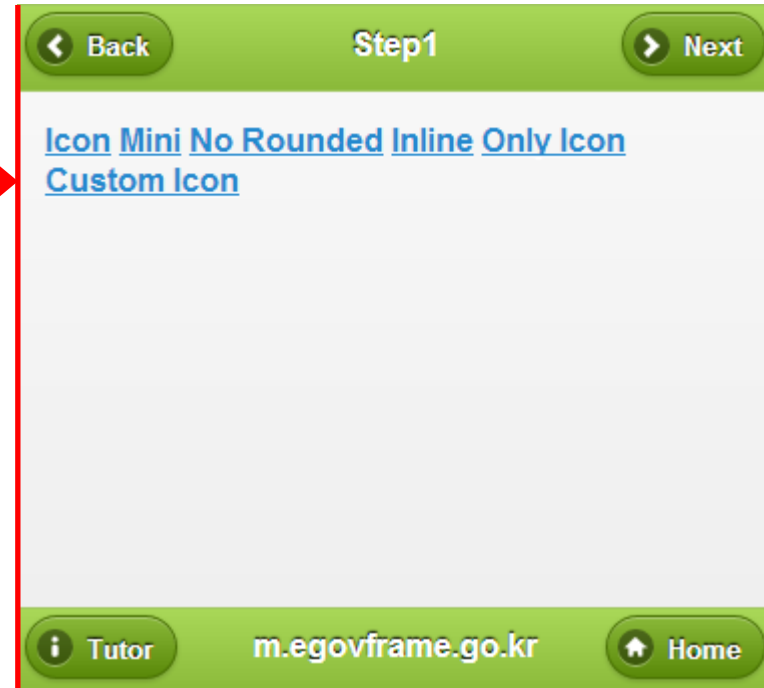
Start.

[Start 03] UX Component 실습 시작하기

- lab402-ux-component를 Tomcat에서 실행 한다.



- Practice Start : 실습 페이지로 이동한다.
- Tutor Page : 해답 페이지로 이동한다.
- Guide : UX Component 가이드 페이지로 이동한다.



- Back : 이전 실습 페이지로 이동한다.
- Next : 다음 실습 페이지로 이동한다.
- Tutor : 해답 페이지로 이동한다.
- Home : 처음으로 돌아간다.

Step1. Button & Icon

[Step 1-01] 기본코드

[Step 1-02] 버튼 색상 변경

[Step 1-03] Icon

[Step 1-04] Custom Icon

[Step 1-05] 버튼 형태 변경

Step1. Button & Icon

파일 경로 : C:/eGovFrame-3.10.0/workspace.edu/lab402-ux-component/src/main/webapp/WEB-INF/jsp/practice/step1.jsp

[Step 1-01] 기본 버튼 생성

- <a> Tag 와 'data-role' 속성을 이용하여 기본 버튼을 생성한다.

```
<a href="#" data-role="button">Icon</a>
<a href="#" data-role="button">Mini</a>
<a href="#" data-role="button">No Rounded</a>
<a href="#" data-role="button">Inline</a>
<a href="#" data-role="button">Only Icon</a>
<a href="#" data-role="button">Custom Icon</a>
```

[Step 1-02] 버튼 색상 변경

- 'data-theme' 속성을 이용하여 버튼의 색상을 변경한다.

```
<a href="#" data-role="button" data-theme="g">Icon</a>
<a href="#" data-role="button" data-theme="f">Mini</a>
<a href="#" data-role="button" data-theme="e">No Rounded</a>
<a href="#" data-role="button" data-theme="c">Inline</a>
<a href="#" data-role="button" data-theme="b">Only Icon</a>
<a href="#" data-role="button" data-theme="a">Custom Icon</a>
```


Step1. Button & Icon

[Step 1-03] Icon 생성

- data-icon 속성을 추가하여 Icon 을 생성 한다.

```
<a href="#" data-role="button" data-theme="g" data-icon="grid">Icon</a>  
<a href="#" data-role="button" data-theme="f" data-icon="star">Mini</a>  
<a href="#" data-role="button" data-theme="e" data-icon="alert">No Rounded</a>  
<a href="#" data-role="button" data-theme="c" data-icon="phone">Inline</a>  
<a href="#" data-role="button" data-theme="b" data-icon="gps">Only Icon</a>  
<a href="#" data-role="button" data-theme="a">Custom Icon</a>
```

- data-iconpos 속성을 추가하여 Icon의 위치를 변경 한다.

```
<a href="#" data-role="button" data-theme="g" data-icon="grid">Icon</a>  
<a href="#" data-role="button" data-theme="f" data-icon="star" data-iconpos="right">Mini</a>  
<a href="#" data-role="button" data-theme="e" data-icon="alert" data-iconpos="top">No Rounded</a>  
<a href="#" data-role="button" data-theme="c" data-icon="phone" data-iconpos="bottom">Inline</a>  
<a href="#" data-role="button" data-theme="b" data-icon="gps" data-iconpos="notext">Only Icon</a>  
<a href="#" data-role="button" data-theme="a">Custom Icon</a>
```

Step1. Button & Icon

[Step 1-04] Custom Icon

- data-icon 속성에 고유한 이름값을 추가한다.

```
<a href="#" data-role="button" data-theme="g" data-icon="grid">Icon</a>
<a href="#" data-role="button" data-theme="f" data-icon="star" data-iconpos="right">Mini</a>
<a href="#" data-role="button" data-theme="e" data-icon="alert" data-iconpos="top">No Rounded</a>
<a href="#" data-role="button" data-theme="c" data-icon="phone" data-iconpos="bottom">Inline</a>
<a href="#" data-role="button" data-theme="b" data-icon="gps" data-iconpos="notext">Only Icon</a>
<a href="#" data-role="button" data-theme="a" data-icon="beer">Custom Icon</a>
```

- CSS에 ui-icon-beer class에 다음과 같이 Custom Icon의 이미지 경로를 정의한다.
 - ※ 실습 JSP파일에 Import css파일을 이용한다. (/src/main/webapp/css/egovframework/mbl/customIcon.css)
 - ※ Custom으로 사용할 이미지는 18x18 픽셀의 사이즈의 투명 배경 PNG-8을 사용해야한다.
- ※ 브라우저에 이미지가 출력되지 않을 경우 (캐시가 갱신되지 않을 경우) 강력한 새로고침을 한다. (윈도우는 Ctrl+Shift+r, 맥은 Command+Shift+r)

```
.ui-icon-beer:after {
    background-image: url(/images/egovframework/mbl/beer.png);
    background-repeat: no-repeat;
}
```

Step1. Button & Icon

[Step 1-05] 버튼 형태 변경

- ❑ `data-mini="true"`를 추가하여 Mini-Button으로 형태를 변경한다.
- ❑ `data-corners="false"` 속성을 이용하여 해당 버튼을 직사각형 형태로 변경한다.
- ❑ `data-inline="true"` 속성을 이용하여 버튼의 넓이를 변경한다.

```
<a href="#" data-role="button" data-theme="g" data-icon="grid">Icon</a>  
<a href="#" data-role="button" data-theme="f" data-icon="star" data-iconpos="right" data-mini="true">Mini</a>  
<a href="#" data-role="button" data-theme="e" data-icon="alert" data-iconpos="top" data-corners="false">No Rounded</a>  
<a href="#" data-role="button" data-theme="c" data-icon="phone" data-iconpos="bottom" data-inline="true">Inline</a>  
<a href="#" data-role="button" data-theme="b" data-icon="gps" data-iconpos="notext">Only Icon</a>  
<a href="#" data-role="button" data-theme="a" data-icon="beer">Custom Icon</a>
```

Step2. Link & Dailog

[Step 2-01] 외부링크 생성

[Step 2-02] 도메인 내부링크 생성

[Step 2-03] 페이지 내부링크 생성

Step2. Link & Dailog

파일 경로 : C:/eGovFrame-3.10.0/workspace.edu/lab402-ux-component/src/main/webapp/WEB-INF/jsp/practice/step2.jsp

[Step 2-01] 외부링크 생성

- <a> Tag를 이용하여 전자정부 표준프레임워크 사이트(" http://m.egovframe.go.kr/mguide/guide/guide.do ")로 이동하는 외부 링크를 생성한다.

```
<a href="http://m.egovframe.go.kr/mguide3.5" data-role="button">외부 링크</a>
```

- Target="home" 속성을 추가하여 가이드 페이지가 새 창에서 열릴 수 있도록 한다.

```
<a href="http://m.egovframe.go.kr/mguide3.5" data-role="button" target="home">외부 링크</a>
```

- 생성된 외부 링크를 클릭하여 해당 링크로의 연결을 확인한다.

Step2. Link & Dailog

[Step 2-02] 도메인 내부링크 생성

- <a> Tag를 이용하여 Home("/internalLinkPage.jsp") 으로 이동하는 도메인 내부링크를 생성한다.

```
<a href="/internalLinkPage.do" data-role="button">도메인 내부 링크</a>
```

- <a> Tag에 'data-transition' 속성 값을 'pop'으로 지정하여 팝업 형식으로 전환효과를 발생시킨다.

```
<a href="/internalLinkPage.do" data-role="button" data-transition="pop">도메인 내부 링크</a>
```

- 내부링크로 연결되는 <a> Tag에 'data-ajax' 속성 값을 'false'로 지정하여 Ajax 방식을 사용하지 않고 페이지를 이동한다.
- Ajax 방식을 사용하지 않고 페이지를 이동하면 각종 페이지 전환효과가 나타나지 않는다.
- 생성된 내부링크를 클릭하여 해당 링크로의 연결과 페이지 전환효과가 나타나지 않는 것을 확인한다.

```
<a href="/internalLinkPage.do" data-role="button" data-ajax="false">도메인 내부 링크</a>
```

Step2. Link & Dailog

[Step 2-03] 페이지 내부링크 버튼 생성

- <a> Tag를 이용하여 페이지 Id가 'sub'인 <div> 페이지(data-role="page")로 이동하는 페이지 내부링크를 생성한다.

```
<a href="#sub" data-role="button">페이지 내부 링크</a>
```

- <div data-role="page"> ~ <div> 영역을 추가하여 내부링크의 타겟 페이지를 생성한다.
'data-role' 속성으로 정의된 페이지에 Id 속성(id="main", id="sub")을 부여하여 각 페이지를 구분한다.

```
<!-- page start -->
<div data-role="page" data-theme="c" id="main">
  :
</div>

<!-- 페이지 내부 이동을 위한 sub page start -->
<div data-role="page" data-theme="e" id="sub">
  :
</div>
```

- '페이지 내부 링크' 버튼의 <a> Tag 의 'data-rel' 속성을 'dialog' 로 지정하여 Dialog 형식으로 페이지를 이동한다.

```
<a href="#sub" data-role="button" data-rel="dialog">페이지 내부 링크</a>
```

- <a> Tag 에 'data-transition' 속성을 추가하여 페이지 전환 효과를 'pop'으로 설정한다.

```
<a href="#sub" data-role="button" data-rel="dialog" data-transition="pop">페이지 내부 링크</a>
```

Step3. Tab, Panel & Panel Widget

[Step 3-01] Tab

[Step 3-02] Panel

[Step 3-03] Panel Widget

Step3. Tab, Panel & Panel Widget

파일 경로 : C:/eGovFrame-3.10.0/workspace.edu/lab402-ux-component/src/main/webapp/WEB-INF/jsp/practice/step3.jsp

[Step 3-01] Tab

- Header에 Tab을 추가한다. class="ui-btn-active"를 통해 활성화 Tab를 설정할 수 있다.

```
<!-- header start -->
<div data-role="header" data-theme="g" id="main">
<a href="/practice/step2.do" data-ajax="false" data-icon="arrow-l">Back</a>
<h1>Step3</h1>
<a href="/practice/step4.do" data-ajax="false" data-icon="arrow-r" class="ui-btn-right">Next</a>

<!-- tab Start -->
<div data-role="tabs">
<ul>
<li><a href="#simpleTab1" class="ui-btn-active">Panel</a></li>
<li><a href="#simpleTab2">Panel Widget</a></li>
</ul>
</div>
<!-- tab end -->
<!-- header end -->
```

- Content영역에 Tab으로 사용할 영역을 추가한다.

```
<!-- content start -->
<div data-role="content">

<div id="simpleTab1">
</div>
<div id="simpleTab2">
</div>

</div>
<!-- content end -->
```

Step3. Tab, Panel & Panel Widget

[Step 3-02] Panel

- ID가 simpleTab1인 영역에 Panel을 추가한다.

```
<!-- content start -->
<div data-role="content">
  <div id="simpleTab1">
    <br>
    <div class="ui-body-g" align="center">
      <h4>Panel</h4>
    </div>
  </div>
  <div id="simpleTab2">
  </div>
</div>
<!-- content end -->
```

Step3. Tab, Panel & Panel Widget

[Step 3-03] Panel Widget

- Page에 Panel Widget을 추가한다.

```
<!-- panel start -->
<div data-role="panel" id="panelWidget">
  <ul data-role="listview" data-theme="g">
    <li><a href="/start.do">Start Page</a></li>
    <li><a href="/tutor/step1.do" data-ajax="false">Step1</a></li>
    <li><a href="/tutor/step2.do" data-ajax="false">Step2</a></li>
    <li><a href="/tutor/step3.do" data-ajax="false">Step3</a></li>
    <li><a href="/tutor/step4.do" data-ajax="false">Step4</a></li>
  </ul>
</div>
<!-- panel end -->
```

- ID가 simpleTab2인 영역에 Panel Widget로 연결할 버튼을 추가한다.

```
<!-- content start -->
<div data-role="content">

  <div id="simpleTab2">
    <a href="#panelWidget" data-role="button">Panel Widget</a>
  </div>

</div>
<!-- content end -->
```

Step3. Tab, Panel & Panel Widget

- ❑ Panel Widget에 닫기버튼을 추가한다.

```
<!-- panel start -->
<div data-role="panel" id="panelWidget">
  <ul data-role="listview" data-theme="g">
    <li><a href="/start.do">Start Page</a></li>
    <li><a href="/tutor/step1.do" data-ajax="false">Step1</a></li>
    <li><a href="/tutor/step2.do" data-ajax="false">Step2</a></li>
    <li><a href="/tutor/step3.do" data-ajax="false">Step3</a></li>
    <li><a href="/tutor/step4.do" data-ajax="false">Step4</a></li>
    <li><a href="#" data-rel="close">Close</a></li>
  </ul>
</div>
<!-- panel end -->
```

- ❑ Panel Widget의 위치를 변경하기 위해 data-position="right"를 추가한다.
- ❑ Panel Widget 디스플레이 방식을 변경하기 위해 data-display="overlay"를 추가한다.
- ❑ Panel Widget 위치 고정을 위해 data-position-fixed="true"를 추가한다.

```
<div data-role="panel" id="panelWidget" data-position="right" data-display="overlay" data-position-fixed="true">
```

Step4. Form, Grid & Popup Widget

[Step 4-01] Text Input Form

[Step 4-02] Slider

[Step 4-03] Grid

[Step 4-04] Popup Widget

Step4. Form, Grid & Popup Widget

파일 경로 : C:/eGovFrame-3.10.0/workspace.edu/lab402-ux-component/src/main/webapp/WEB-INF/jsp/practice/step4.jsp

[Step 4-01] Text Input Form

- 화면 가로 길이에 맞는 폼을 구성하기 위해 data-role="fieldcontain" 속성을 가진 <div> 태그를 추가한다.

```
<div data-role="fieldcontain">
  <label>Text input:</label>
  <input type="text" name="text-basic" id="text-basic" value="">
</div>
```

- <label> 태그에 <input>의 id값을 가지는 for 속성을 추가한다.

```
<div data-role="fieldcontain">
  <label for="text-basic">Text input:</label>
  <input type="text" name="text-basic" id="text-basic" value="">
</div>
```

Step4. Form, Grid & Popup Widget

- ❑ <Input> 태그에 placeholder 속성을 추가하여 Form의 설명을 추가한다.

```
<div data-role="fieldcontain">
  <label for="text-basic">Text input:</label>
  <input type="text" name="text-basic" id="text-basic" value="" placeholder="Text input">
</div>
```

- ❑ <label> 영역을 화면에 나타내지 않으려면 태그를 추가한다.

```
<div data-role="fieldcontain">
  <label for="text-basic" class="ui-hidden-accessible">Text input:</label>
  <input type="text" name="text-basic" id="text-basic" value="" placeholder="Text input">
</div>
```

- ❑ 입력을 받지 않는 Form을 사용하기 위해서는 disabled="disabled" 속성을 추가한다.

```
<div data-role="fieldcontain">
  <label for="text-basic" class="ui-hidden-accessible">Text input:</label>
  <input type="text" name="text-basic" id="text-basic" value="" placeholder="Text input" disabled="disabled">
</div>
```

Step4. Form, Grid & Popup Widget

[Step 4-02] Slider

- 증감 단위를 설정하려면 **step** 속성을 확인한다.

```
<label for="slidersingle">Rangeslider:</label>  
<input type="range" name="slidersingle" id="slidersingle" min="0" max="10" value="2.5" step=".1">
```

- **data-theme** 속성을 추가하여 Slider 색상을 변경한다.

```
<label for="slidersingle">Rangeslider:</label>  
<input type="range" name="slidersingle" id="slidersingle" min="0" max="10" value="2.5" step=".1" data-theme="f">
```

- Slider의 High Light 부분과 배경 트랙부분 색상을 변경하기 위해 **data-highlight="true"**와 **data-track-theme="g"**를 추가한다.

```
<label for="slidersingle">Rangeslider:</label>  
<input type="range" name="slidersingle" id="slidersingle" min="0" max="10" step=".1" value="2.5" data-theme="f"  
data-highlight="true" data-track-theme="g">
```

- 두 개의 Slider를 **<div data-role="rangeslider"> ~ </div>**로 묶으면 Range Slider로 사용할 수 있다.

```
<div data-role="rangeslider">  
  <label for="range-a">Rangeslider:</label>  
  <input type="range" name="range-a" id="range-a" min="0" max="10" step=".1" value="2.5">  
  <label for="range-b">Rangeslider:</label>  
  <input type="range" name="range-b" id="range-b" min="0" max="10" step=".1" value="7.5">  
</div>
```


Step4. Form, Grid & Popup Widget

[Step 4-03] Grid

- 두 줄로 화면에 표시되고 있는 슬라이더에 그리드를 적용하여 한 라인에 출력한다.

```
<div class="ui-grid-a">
  <div class="ui-block-a">
    <label for="slidersingle">Rangeslider:</label>
    <input type="range" name="slidersingle" id="slidersingle" min="0" max="10" value="2.5" step=".1" data-theme="f"
      data-highlight="true" data-track-theme="g">
  </div>
  <div class="ui-block-b">
    <div data-role="rangeslider">
      <label for="range-a">Rangeslider:</label>
      <input type="range" name="range-a" id="range-a" min="0" max="10" value="2.5" step=".1">
      <label for="range-b">Rangeslider:</label>
      <input type="range" name="range-b" id="range-b" min="0" max="10" value="7.5" step=".1">
    </div>
  </div>
</div>
```

Step4. Form, Grid & Popup Widget

- 각 그리드 영역에 Padding을 주려면 <div class="ui-bar"> ~ </div/>를 추가한다.
ui-bar-[data-theme]를 추가하면 각 테마의 색상을 그리드의 배경색으로 사용할 수 있다.

```
<div class="ui-grid-a">
  <div class="ui-block-a">
    <div class="ui-bar ui-bar-e">
      <label for="slidersingle">Rangeslider:</label>
      <input type="range" name="slidersingle" id="slidersingle" min="0" max="10" value="2.5" step=".1" data-theme="f"
        data-highlight="true" data-track-theme="g">
    </div>
  </div>
  <div class="ui-block-b">
    <div class="ui-bar ui-bar-e">
      <div data-role="rangeslider">
        <label for="range-a">Rangeslider:</label>
        <input type="range" name="range-a" id="range-a" min="0" max="10" value="2.5" step=".1">
        <label for="range-b">Rangeslider:</label>
        <input type="range" name="range-b" id="range-b" min="0" max="10" value="7.5" step=".1">
      </div>
    </div>
  </div>
</div>
```

Step4. Form, Grid & Popup Widget

- 그리드와 같은 Padding을 가지는 다른 하나의 영역을 사용하려면 그리드로 영역 아래에 `<div class="ui-grid-solo"> ~ </div/>`를 추가한다.

```
<div class="ui-grid-a">
<div class="ui-block-a">
<div class="ui-bar ui-bar-e">
<label for="slidersingle">RangeSlider:</label>
<input type="range" name="slidersingle" id="slidersingle" min="0" max="10" value="2.5" step=".1" data-theme="f"
data-highlight="true" data-track-theme="g">
</div>
</div>
<div class="ui-block-b">
<div class="ui-bar ui-bar-e">
<div data-role="rangeslider">
<label for="range-a">RangeSlider:</label>
<input type="range" name="range-a" id="range-a" min="0" max="10" value="2.5" step=".1">
<label for="range-b">RangeSlider:</label>
<input type="range" name="range-b" id="range-b" min="0" max="10" value="7.5" step=".1">
</div>
</div>
</div>
</div>

<div class="ui-grid-solo">
<div class="ui-block-a">
<button type="button" data-theme="b">Grid Solo</button>
</div>
</div>
```

Step4. Form, Grid & Popup Widget

[Step 4-04] Popup Widget

- Popup Widget을 Content 영역에 추가한다.

```
<div data-role="popup" id="popupBasic" class="ui-content">  
  <p>Popup Widget 입니다.</p>  
</div>
```

- Popup Widget 링크 버튼에 data-rel="popup" 속성을 추가하고, href 값으로 Popup Widget의 ID를 추가한다.

```
<a href="#popupBasic" data-rel="popup" data-role="button">Popup Widget</a>
```

- 링크 버튼에 data-transition=" [none, pop, fade, flip, turn, flow, slide, slidefade, slideup, slidedown]" 속성을 추가하여 팝업 표시 효과를 변경한다.

```
<a href="#popupBasic" data-rel="popup" data-role="button" data-transition="fade">Popup Widget</a>
```

Step4. Form, Grid & Popup Widget

- Popup Widget에 닫기 버튼을 추가한다. class= "[ui-btn-right, ui-btn-left]"를 사용하여 버튼의 위치를 지정할 수 있다.

```
<div data-role="popup" id="popupBasic" class="ui-content">
  <a href="#" data-rel="back" data-role="button" data-theme="a" data-icon="delete" data-iconpos="notext"
    class="ui-btn-left">Close</a>
  <p>Popup Widget 입니다.</p>
</div>
```

- Popup Widget에 data-dismissible="false" 추가 하면 닫기 버튼 이외에 입력에 Popup Widget이 닫히지 않는다.

```
<div data-role="popup" id="popupBasic" class="ui-content" data-dismissible="false">
  <a href="#" data-rel="back" data-role="button" data-theme="a" data-icon="delete" data-iconpos="notext"
    class="ui-btn-left">Close</a>
  <p>Popup Widget 입니다.</p>
</div>
```

- data-theme 속성을 추가하여 Popup Widget의 색상을 설정한다.
data-overlay-theme 속성을 추가하여 Popup 창 외부의 Overlay영역의 색상을 지정할 수 있다.

```
<div data-role="popup" id="popupBasic" class="ui-content" data-dismissible="false" data-theme="e" data-overlay-theme="a">
  <a href="#" data-rel="back" data-role="button" data-theme="a" data-icon="delete" data-iconpos="notext"
    class="ui-btn-left">Close</a>
  <p>Popup Widget 입니다.</p>
</div>
```