

전자정부 모바일 표준프레임워크
실행 환경 컴포넌트 별 실습



UX Component 실습 목차

- Step 1. Page Structure
- Step 2. Button
- Step 3. Panel
- Step 4. Internal / External Link
- Step 5. Label / Text
- Step 6. Tabs
- Step 7. Form
- Step 8. Menu
- Step 9. Progress Dialog / Bar
- Step 10. Dialog
- Step 11. Grid View
- Step 12. List View
- Step 13. Date / Time Picker
- Step 14. Check / Radio
- Step 15. Icon
- Step 16. Selector / Switch

Step1. Page Structure

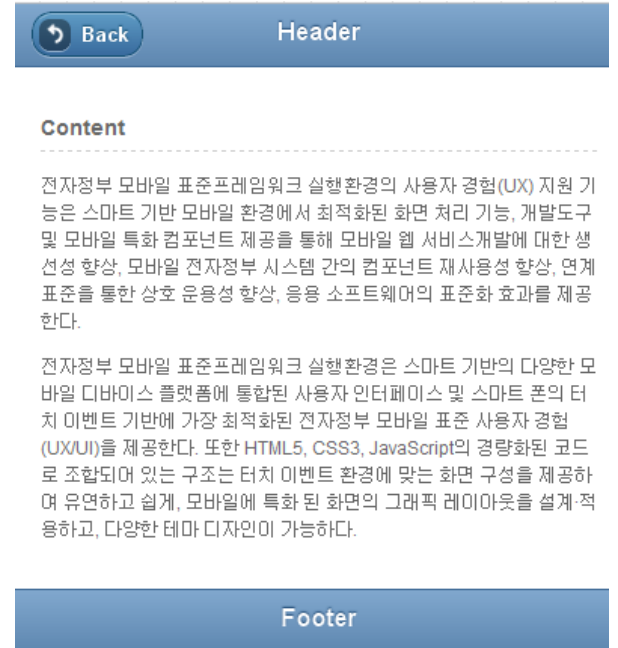
[Step 1-01] 'viewport' 설정

[Step 1-02] Library Import

[Step 1-03] <body>영역 페이지 구조

[Step 1-04] 테마 적용

[Step 1-05] Header 영역에 버튼 추가



Step1. Page Structure

[Step 1-01] '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">
```

[Step 1-02] Library Import

- 전자정부 모바일 표준프레임워크 실행환경을 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="/css/egovframework/mbl/cmm/jquery.mobile-1.0b3.min.css" />  
<link rel="stylesheet" href="/css/egovframework/mbl/cmm/EgovMobile.css" />  
<script type="text/javascript" src="/js/egovframework/mbl/cmm/jquery-1.6.2.min.js"></script>  
<script type="text/javascript" src="/js/egovframework/mbl/cmm/jquery.mobile-1.0b3.min.js"></script>  
<script type="text/javascript" src="/js/egovframework/mbl/cmm/EgovMobile.js"></script>
```

Step1. Page Structure

[Step 1-03] <body>영역 페이지 구조

- 전자정부 모바일 표준프레임워크 실행 환경의 사용자 경험(UX) 지원 기능을 효과적으로 적용하기 위해서
- <body>를 영역별로 구분한다.
- <body>는 'data-role' 속성에 적용된 값에 따라 Page, Header, Content, Footer 영역으로 구분된다.

```
<body>
  <div data-role="page">
    <div data-role="header">
      <h1>Header</h1>
    </div>
    <div data-role="content">
      <p>Content</p>
    </div>
    <div data-role="footer">
      <h4>Footer</h4>
    </div>
  </div>
</body>
```

Step1. Page Structure

[Step 1-04] 테마 적용

- 'data-theme' 속성을 이용하여 Page 영역에 'd'테마(흰색)를, Header 와 Footer 영역에 'b'테마를 적용한다.

```
<div data-role="page" data-theme="d">
  <div data-role="header" data-theme="b">
    <h1>Header</h1>
  </div>

  <div data-role="content">
    <p>Content</p>
  </div>

  <div data-role="footer" data-theme="b">
    <h4>Footer</h4>
  </div>
</div>
```

[Step 1-05] Header 영역에 버튼 추가

- <a> Tag 의 'data-rel' 속성 값을 'back' 으로 지정하여 Header 영역에 이전 페이지로 이동하는 버튼을 생성한다.
- 'data-icon' 속성 값을 지정하여 뒤로가기 모양(back)의 아이콘을 추가한다.

```
<a data-rel="back" data-icon="back">Back</a>
```

Step2. Button

[Step 2-01] 기본 버튼 생성

[Step 2-02] 인라인 버튼 생성

[Step 2-03] 버튼 그룹 생성

[Step 2-04] 버튼 색상 변경



기본 버튼 생성



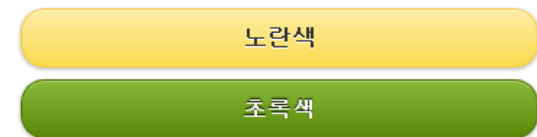
인라인 버튼 생성



버튼 그룹 생성



버튼 색상 변경



Step2. Button

[Step 2-01] 기본 버튼 생성

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

```
<a data-role="button">기본 버튼</a>
```

[Step 2-02] 인라인 버튼 생성

- 'data-inline' 속성을 이용하여 버튼의 너비가 텍스트에 맞춰진 인라인 버튼을 생성한다.

```
<a data-role="button" data-inline="true">인라인 버튼</a>
```

[Step 2-03] 버튼 그룹 생성

- 'data-role' 속성을 갖는 <div> Tag를 이용하여 3개의 버튼이 세로로 배열된 버튼 그룹을 생성한다.
- 'controlgroup' 을 'data-role' 속성 값으로 갖는 <div> Tag 를 이용하여 버튼 그룹을 생성할 수 있다.

```
<div data-role="controlgroup">  
  <a data-role="button">버튼 그룹 1</a>  
  <a data-role="button">버튼 그룹 2</a>  
  <a data-role="button">버튼 그룹 3</a>  
</div>
```


Step2. Button

[Step 2-04] 버튼 색상 변경

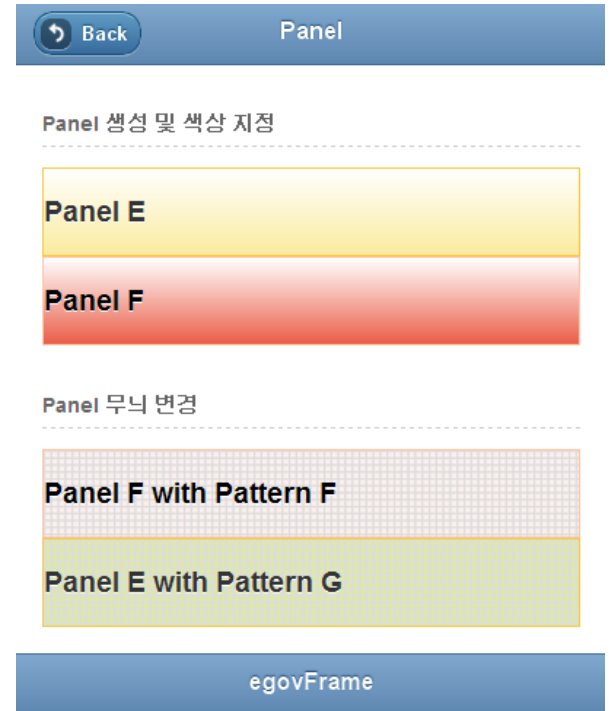
- 'data-theme' 속성을 이용하여 'e'테마(노란색)가 적용된 버튼과 'g'테마(초록색)가 적용된 버튼을 생성한다.

```
<a data-role="button" data-theme="e">노란색</a>  
<a data-role="button" data-theme="g">초록색</a>
```

Step3. Panel

[Step 3-01] Panel 생성 및 색상 지정

[Step 3-02] Panel 무늬 변경



Step3. Panel

[Step 3-01] Panel 생성 및 색상 지정

- ❑ <div> Tag 에 class 속성을 지정하여 'e'테마가 적용된 Panel 을 생성한다.
- ❑ <div> Tag 에 class 속성을 지정하여 'f'테마가 적용된 Panel 을 생성한다.
- ❑ Panel의 class 속성 값을 'ui-body-[a~g]'으로 지정하여 Panel에 색상을 적용할 수 있다.
- ❑ Panel 내부의 텍스트 크기는 <h3> 으로 지정한다.

```
<!-- 'e' 테마 Panel -->  
<div class="ui-body-e">  
  <h3>Panel E</h3>  
</div>  
  
<!-- 'f' 테마 Panel -->  
<div class="ui-body-f">  
  <h3>Panel F</h3>  
</div>
```

Step3. Panel

[Step 3-02] Panel 무늬 변경

- ❑ <div> Tag 에 class 속성을 추가하여 'f'테마가 적용된 Panel에 'f'테마의 무늬를 적용한다.
- ❑ <div> Tag 에 class 속성을 추가하여 'e'테마가 적용된 Panel에 'g'테마의 무늬를 적용한다.
- ❑ Panel의 class 속성 값에 'egov-panel-pattern-[a~g]' 을 추가하여 Panel에 무늬를 적용할 수 있다.
- ❑ Panel 내부의 텍스트 크기는 <h3> 으로 지정한다.

```
<!-- 'f' 테마 'f' 무늬 -->
<div class="ui-body-f egov-panel-pattern-f">
  <h3>Panel F with Pattern F</h3>
</div>

<!-- 'e' 테마 'g' 무늬 -->
<div class="ui-body-e egov-panel-pattern-g">
  <h3>Panel E with Pattern G</h3>
</div>
```

Step4. Internal / External Link

[Step 4-01] 외부링크 생성

[Step 4-02] 도메인 내부링크 생성

Back Link

외부링크 생성

[모바일 실행환경 가이드](#)

도메인 내부링크 생성

[Home](#)

Ajax 사용 해제

[Home](#)

페이지 내부링크 버튼 생성

Button for Page 2

egovFrame

Step4. Internal / External Link

[Step 4-01] 외부링크 생성

- ❑ <a> Tag를 이용하여 모바일 실행환경 가이드 사이트(<http://mguide.egovframe.go.kr/>)로 이동하는 외부 링크를 생성한다.
- ❑ 생성된 외부 링크를 클릭하여 해당 링크로의 연결을 확인한다.

```
<a href="http://mguide.egovframe.go.kr/">모바일 실행환경 가이드</a>
```

[Step 4-02] 도메인 내부링크 생성

- ❑ <a> Tag를 이용하여 Home("/sample/home.do") 으로 이동하는 도메인 내부링크를 생성한다.
- ❑ 생성된 도메인 내부링크를 클릭하여 해당 링크로의 연결과 각종 페이지 전환효과를 확인한다.

```
<a href="/sample/home.do">Home</a>
```

[Step 4-03] Ajax 사용 해제

- ❑ 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="/sample/home.do" data-ajax="false">Home</a>
```

Step4. Internal / External Link

[Step 4-04-1] 페이지 내부링크 버튼 생성

- ❑ <a> Tag를 이용하여 페이지 Id가 'page2'인 <div> 페이지(data-role="page")로 이동하는 페이지 내부링크를 생성한다.
- ❑ <a> Tag에 'data-role' 속성을 추가하여 버튼 형태의 링크를 생성한다.

```
<a href="#page2" data-role="button">Button for Page 2</a>
```

[Step 4-04-2] 페이지 내부링크 버튼 생성

- ❑ 'data-role' 속성으로 정의된 페이지에 Id 속성을 부여하여 페이지를 구분한다.
- ❑ [Step 4-04-1] 에서 생성한 페이지 내부링크 버튼을 클릭하여 해당 링크로의 연결을 확인한다.

```
<div id="page2" data-role="page" data-theme="d">
```

Step5. Label / Text

[Step 5-01] 텍스트 정렬

[Step 5-02] 폰트 설정

[Step 5-03] 텍스트 색상 변경

← Back Label / Text

텍스트 정렬

왼쪽 정렬

가운데 정렬

오른쪽 정렬

폰트 설정

tahoma Font

verdana Font

텍스트 색상 변경

초록색

빨간색

파란색

egovFrame

Step5. Label / Text

[Step 5-01] 텍스트 정렬

- ❑ <div> Tag 에 class 속성을 지정하여 텍스트를 왼쪽 정렬한다.
- ❑ <div> Tag 에 class 속성을 지정하여 텍스트를 가운데 정렬한다.
- ❑ <div> Tag 에 class 속성을 지정하여 텍스트를 오른쪽 정렬한다.
- ❑ class 속성 값을 'egov-align-[left, center, right]'로 지정하여 텍스트를 정렬할 수 있다.

```
<div class="egov-align-left"><h3>왼쪽 정렬</h3></div>  
<div class="egov-align-center"><h3>가운데 정렬</h3></div>  
<div class="egov-align-right"><h3>오른쪽 정렬</h3></div>
```

[Step 5-02] 폰트 설정

- ❑ <div> Tag 에 class 속성을 지정하여 'egov-font-tahoma' 으로 텍스트의 폰트를 변경한다.
- ❑ <div> Tag 에 class 속성을 지정하여 'egov-font-verdana' 으로 텍스트의 폰트를 변경한다.

```
<div class="egov-font-tahoma"><h3>tahoma Font</h3></div>  
<div class="egov-font-verdana"><h3>verdana Font</h3></div>
```

Step5. Label / Text

[Step 5-03] 텍스트 색상 변경

- ❑ <div> Tag 에 class 속성을 지정하여 텍스트를 초록색으로 변경한다.
- ❑ <div> Tag 에 class 속성을 지정하여 텍스트를 빨간색으로 변경한다.
- ❑ <div> Tag 에 class 속성을 지정하여 텍스트를 파란색으로 변경한다.
- ❑ class 속성 값을 'egov-color-[green, red, blue]'로 지정하여 텍스트의 색상을 변경할 수 있다.

```
<div class="egov-color-green"><h3>초록색</h3></div>  
<div class="egov-color-red"><h3>빨간색</h3></div>  
<div class="egov-color-blue"><h3>파란색</h3></div>
```

Step6. Tabs

[Step 6-01] Header Tab 생성

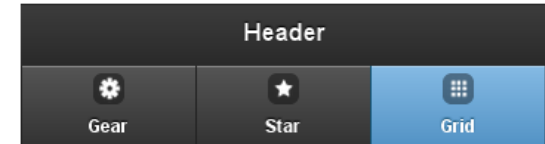
[Step 6-02] 탭 아이콘 추가



Header Tab 생성



탭 아이콘 추가



Step6. Tabs

[Step 6-01] Header Tab 생성

- Header 영역에 'navbar'를 'data-role' 속성 값으로 갖는 Tab을 생성한다.
- Tab이 2개의 항목을 갖고 1번 째 항목이 기본적으로 활성화되도록 설정한다.
- Tab 첫 번째 항목의 <a> Tag에 Home("/sample/home.do")으로 이동하는 링크를 추가한다.
- Tab 두 번째 항목의 <a> Tag에 Page Structure("/sample/page.do")으로 이동하는 링크를 추가한다.

```
<div data-role="header">
  <h1>Header</h1>
  <div data-role="navbar">
    <ul>
      <li><a href="/sample/home.do" class="ui-btn-active">Home</a></li>
      <li><a href="/sample/page.do">Page Structure</a></li>
    </ul>
  </div>
</div>
```

Step6. Tabs

[Step 6-02] 탭 아이콘 추가

- Tab 항목의 <a> Tag에 'data-icon' 속성을 적용하여 순서대로 Gear, Star, Grid 형태의 아이콘을 Tab에 추가한다.
- Tab이 3개의 항목을 갖고 3번 째 항목이 기본적으로 활성화되도록 설정한다.

```
<div data-role="navbar">  
  <ul>  
    <li><a data-icon="gear">Gear</a></li>  
    <li><a data-icon="star">Star</a></li>  
    <li><a data-icon="grid" class="ui-btn-active">Grid</a></li>  
  </ul>  
</div>
```

Step7. Form

[Step 7-01] 기본 폼 배열

[Step 7-02] 폼 구성요소

Back Form

기본 폼 배열

Text Input :

폼 구성요소

Text Input :

Search Input :

Slider :

Submit Button

egovFrame

Step7. Form

[Step 7-01] 기본 폼 배열

- ❑ 폼 구성요소들을 'data-role' 속성 값이 'fieldcontain'인 <div> Tag 내부에 위치시켜 기본 배열이 갖춰진 폼을 생성한다.
- ❑ <label> Tag 의 'for' 속성 값과 폼 구성요소의 'id' 속성 값을 동일하게 부여하여 Label 과 폼 구성요소가 짝을 이룰 수 있도록 한다.
- ❑ Text Input 한 개를 구성요소로 갖는 폼을 생성한다.
- ❑ Label 로 해당 Text Input 에 대한 타이틀을 명시한다.

```
<form>  
  <div data-role="fieldcontain">  
    <label for="textInput">Text Input : </label>  
    <input id="textInput" type="text" />  
  </div>  
</form>
```

Step7. Form

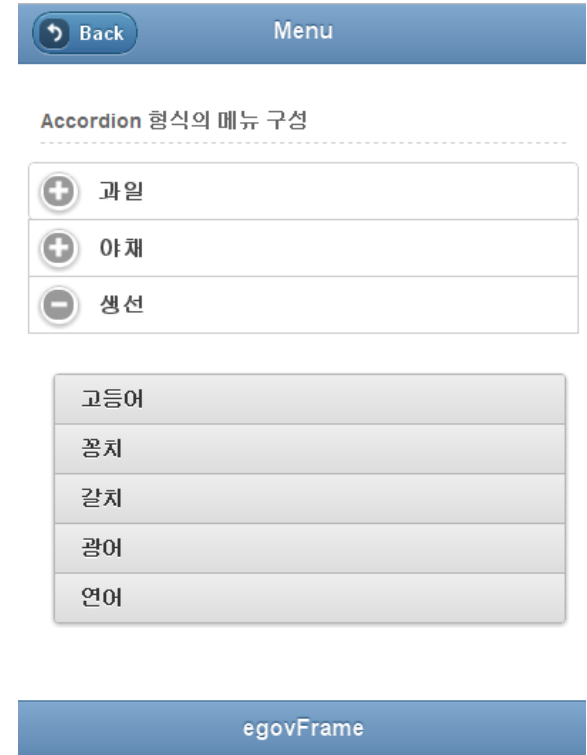
[Step 7-02] 폼 구성요소

- 폼 구성요소의 종류는 Text Inputs, Search Inputs, Sliders, Switches, Radio Buttons, Check Boxes, Selectors 가 있다.
- 기본 폼 배열에 Text Input, Search Input, Slider, Form Button 를 폼 구성요소로 추가한다.
- <input> Tag 의 'type' 속성 값을 'search'로 지정하여 Search Input 을 생성할 수 있다.
- <input>Tag 의 'type' 속성 값을 'range'로 지정하여 Slider 를 생성할 수 있다.
- Slider 는 'min', 'max' 속성을 추가로 지정해주어야 한다.
- <button> Tag 의 'type' 속성을 'submit'으로 지정하여 Submit Button 을 생성한다.

```
<form>
  <div data-role="fieldcontain">
    <label for="text">Text Input : </label>
    <input id="text" type="text" />
  </div>
  <div data-role="fieldcontain">
    <label for="search">Search Input : </label>
    <input id="search" type="search" />
  </div>
  <div data-role="fieldcontain">
    <label for="slider">Slider : </label>
    <input id="slider" type="range" min="0" max="100" />
  </div>
  <div data-role="fieldcontain">
    <button type="submit" id="submit">Submit Button</button>
  </div>
</form>
```


Step8. Menu

[Step 8-01] Accordion 형식의 메뉴 구성



Step8. Menu

[Step 8-01] Accordion 형식의 메뉴 구성

- 모바일 페이지의 메뉴 기능은 **Dialog, Grid, List, Collapsible** 의 기능을 활용하여 구현할 수 있다.
- Collapsible** 기능을 이용하여 **Accordion** 형식의 메뉴를 구성한다.
- '**data-role**' 속성 값이 '**collapsible-set**'인 **<div> Tag** 내부에 '**data-role**' 속성 값이 '**collapsible**'인 **<div> Tag**를 추가하여 상위메뉴를 생성한다.
- 상위메뉴의 **<div> Tag**에 리스트를 추가하여 하위메뉴를 구성한다.
- 상위메뉴의 텍스트는 **<h3>**으로 지정한다.**<form>**

Step8. Menu

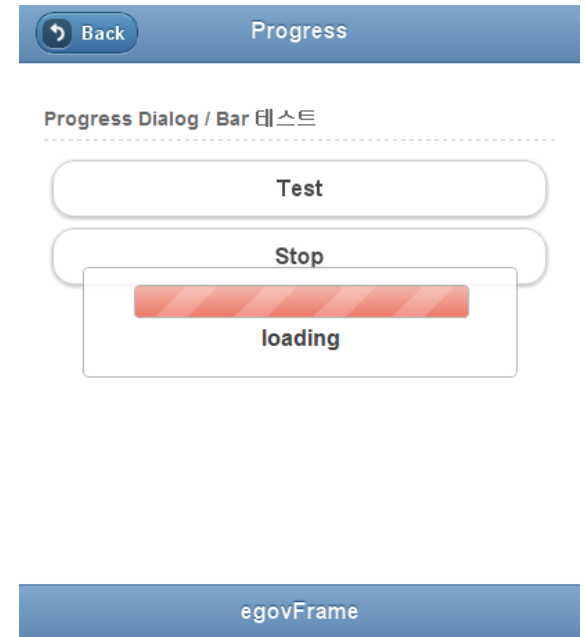
[Step 8-01] Accordion 형식의 메뉴 구성

```
<div data-role="collapsible-set">
  <div data-role="collapsible">
    <h4>과일</h4>
    <ul data-role="listview" data-inset="true">
      <li>사과</li>
      <li>바나나</li>
      <li>포도</li>
      <li>딸기</li>
      <li>수박</li>
    </ul>
  </div>
  <div data-role="collapsible">
    <h4>야채</h4>
    <ul data-role="listview" data-inset="true">
      <li>토마토</li>
      <li>오이</li>
      <li>배추</li>
      <li>양배추</li>
      <li>양파</li>
    </ul>
  </div>
  <div data-role="collapsible">
    <h4>생선</h4>
    <ul data-role="listview" data-inset="true">
      <li>고등어</li>
      <li>꽁치</li>
      <li>갈치</li>
      <li>광어</li>
      <li>연어</li>
    </ul>
  </div>
</div>
```

Step9. Progress Dialog / Bar

[Step 9-01] Progress Bar 설정

[Step 9-02] Progress Dialog / Bar 색상 변경



Step9. Progress Dialog / Bar

[Step 9-01] Progress Bar 설정

- 기본적으로 Progress Dialog 가 나타나는 설정을 JavaScript 코드를 추가하여 Progress Bar 로 변경한다.
- Test 버튼을 클릭하여 Progress Bar 가 기본적으로 나타나는 지 확인한다.

```
<script type="text/javascript" >
    $(document).bind("mobileinit", function(){
        $.mobile.progressBar = true;
    });
</script>
```

[Step 9-02] Progress Dialog / Bar 색상 변경

- JavaScript 코드의 \$.mobile.progressTheme 값을 'f'로 지정하여 빨간색으로 색상을 변경한다.
- Test 버튼을 클릭하여 Progress Dialog / Bar 의 색상이 빨간색으로 나타나는 지 확인한다.

```
<script type="text/javascript" >
    $(document).bind("mobileinit", function(){
        $.mobile.progressTheme = 'f';
    });
</script>
```

Step10. Dialog

[Step 10-01] Dialog 페이지 이동

[Step 10-02-1] Action Sheet 생성

[Step 10-02-2] Action Sheet 생성

[Step 10-03-1] Alert Dialog 생성

[Step 10-03-2] Alert Dialog 생성

Action sheet Dialog

선택하세요!

선택1

선택2

닫기

Dialog

Back

페이지 이동

Home

Action Sheet 생성

You Entered : 선택2

Action Sheet

Alert Dialog 생성

Alert Dialog

egovFrame

Step10. Dialog

[Step 10-01] Dialog 페이지 이동

- ❑ <a> Tag 의 'data-rel' 속성을 'dialog' 로 지정하여 Dialog 형식으로 페이지를 이동한다.
- ❑ 버튼 형식으로 Home("/sample/home.do") 으로 이동하는 Dialog 링크를 생성한다.
- ❑ <a> Tag 에 'data-transition' 속성을 추가하여 페이지 전환 효과를 'flip' 으로 설정한다.

```
<a href="/sample/home.do" data-rel="dialog" data-transition="flip" data-role="button">Home</a>
```

[Step 10-02-1] Action Sheet 생성

- ❑ JavaScript 코드를 추가하여 Action Sheet 를 생성한다.
- ❑ btmItem 변수를 수정하여 2개의 선택 항목을 갖고 각각의 값이 '선택1', '선택2' 인 Action Sheet 를 생성한다.
- ❑ jActionSheet 메소드의 첫 번째 인자 값을 변경하여 '선택하세요!' 라는 메시지가 나타나도록 설정한다.

```
<script type="text/javascript">
    $(document).ready( function() {
        $("#actionsheet_dialog").click( function() {
            var btmItem = [{id : 'button1', value: "선택1"},
                {id : 'button2', value: "선택2"}];
            jActionSheet('선택하세요!', 'Action sheet Dialog', 'a', btmItem , function(r) {
                $('#actionsheetOut').text(r);
            });
        });
    });
</script>
```

Step10. Dialog

[Step 10-02-2] Action Sheet 생성

- 'button' type 의 <input> Tag 에 id 속성 값을 'actionsheet_dialog'로 지정하여 Action Sheet 를 호출할 수 있도록 한다.
- id 속성 값이 'actionsheetOut'인 Tag 를 추가하여 선택된 값이 화면에 출력될 수 있도록 한다.
- 버튼을 클릭하여 Action Sheet 가 나타나는 지 확인한다.

```
<label for="actionsheet_dialog" class="center">You Entered : <strong id="actionsheetOut"></strong></label>
<input id="actionsheet_dialog" type="button" value="Action Sheet" />
```

[Step 10-03-1] Alert Dialog 생성

- JavaScript 코드를 추가하여 Alert Dialog 를 생성한다.
- jAlert 메소드의 첫 번째 인자 값을 변경하여 '주의하세요!' 라는 메시지가 나타나도록 설정한다.

```
<script type="text/javascript">
    $(document).ready( function() {
        $("#alert_dialog").click( function() {
            jAlert('주의하세요!', 'Alert Dialog', 'a');
        });
    });
</script>
```


Step10. Dialog

[Step 10-03-2] Alert Dialog 생성

- ❑ 'button' type 의 <input> Tag 에 id 속성 값을 'alert_dialog'로 지정하여 Alert Dialog 를 호출할 수 있도록 한다.
- ❑ 버튼을 클릭하여 Alert Dialog 가 나타나는 지 확인한다.

```
<input id="alert_dialog" type="button" value="Alert Dialog" />
```

Step11. Grid View

[Step 11-01] 기본 Grid View 생성

[Step 11-02] 다중 행 Grid View 생성

[Step 11-03] Grid View 버튼 배열

Back Grid View

기본 Grid View 생성

A1 B1 C1

다중 행 Grid View 생성

A1	B1
A2	B2

Grid View 버튼 배열

A1	B1	C1
A2	B2	C2

egovFrame

Step11. Grid View

[Step 11-01] 기본 Grid View 생성

- <div> Tag 에 class 속성을 추가하여 3개의 셀을 갖고 화면을 3등분하는 Grid View 를 생성한다.

```
<div class="ui-grid-b">  
  <div class="ui-block-a">A1</div>  
  <div class="ui-block-b">B1</div>  
  <div class="ui-block-c">C1</div>  
</div>
```

Step11. Grid View

[Step 11-02] 다중 행 Grid View 생성

- ❑ Grid View의 셀 내용에 Panel 을 적용하여 Panel 이 배경으로 적용된 Grid View 를 생성한다.
- ❑ 2개의 행을 갖고 화면을 2등분하는 Grid View 를 생성한다.
- ❑ class 속성 값이 "ui-grid-" 으로 시작하는 <div> Tag 내부에
- ❑ class 속성 값이 "ui-block-" 으로 시작하는 <div> Tag 를 반복적으로 배치하여 다중행 Grid View 를 생성한다.
- ❑ 첫 번째 행에 'a'테마가 적용된 Panel 을 추가하여 검정색으로 지정한다.
- ❑ 두 번째 행에 'g'테마가 적용된 Panel 을 추가하여 초록색으로 지정한다.

```
<div class="ui-grid-a">
  <div class="ui-block-a"><div class="ui-body-f egov-align-center"><h3>A1</h3></div></div>
  <div class="ui-block-b"><div class="ui-body-f egov-align-center"><h3>B1</h3></div></div>
  <div class="ui-block-a"><div class="ui-body-e egov-align-center"><h3>A2</h3></div></div>
  <div class="ui-block-b"><div class="ui-body-e egov-align-center"><h3>B2</h3></div></div>
</div>
```

Step11. Grid View

[Step 11-03] Grid View 버튼 배열

- Grid 선언 클래스 내 컴포넌트를 추가하면 해당 컴포넌트가 그 크기에 맞게 나타난다.
- Grid View 의 셀 내용으로 버튼을 추가하여 화면을 4등분 하고 2개의 행을 갖는 Grid View 버튼 배열을 생성한다.

```
<div class="ui-grid-b">  
  <div class="ui-block-a"><a data-role="button">A1</a></div>  
  <div class="ui-block-b"><a data-role="button">B1</a></div>  
  <div class="ui-block-c"><a data-role="button">C1</a></div>  
  
  <div class="ui-block-a"><a data-role="button">A2</a></div>  
  <div class="ui-block-b"><a data-role="button">B2</a></div>  
  <div class="ui-block-c"><a data-role="button">C2</a></div>  
</div>
```

Step12. List View

[Step 12-01] 기본 List View 생성 (Read Only)

[Step 12-02] 기본 List View 생성 (Link)

[Step 12-03] List Content Formatting

[Step 12-04] Split Button & Search Filter 생성

[Step 12-05] Nested List 생성

Back List View

기본 List View 생성 (Read Only)

Home
Page Structure
List View

기본 List View 생성 (Link)

Home	▶
Page Structure	▶
List View	▶

List Content Formatting

과일	
사과 무기질이 풍부한 과일	apple ▶
바나나 탄수화물이 풍부한 과일	banana ▶
야채	
오이 수분이 많은 야채	▶
양배추 위궤양에 도움이 되는 야채	▶
당근 지용성 비타민이 풍부한 야채	▶

Split Button & Search Filter 생성

Filter items...

Home	▶
Page Structure	▶
List View	▶

Nested List 생성

과일	▶
야채	▶

egovFrame

Step12. List View

[Step 12-01] 기본 List View 생성 (Read Only)

- ❑ Tag 에 'data-role' 속성을 추가하여 기본 List View 를 생성한다.
- ❑ Tag 에 'data-inset' 속성을 추가하여 화면 레이아웃에 알맞는 List View 를 생성한다.
- ❑ 'Home', 'Page Structure', 'List View' 를 항목으로 갖는 List View 를 생성한다.

```
<ul data-role="listview" data-inset="true">
  <li>Home</li>
  <li>Page Structure</li>
  <li>List View</li>
</ul>
```

[Step 12-02] 기본 List View 생성 (Link)

- ❑ 'Home', 'Page Structure', 'List View' 를 항목으로 갖는 List View 를 생성한다.
- ❑ 리스트 항목(Tag) 에 <a> Tag 를 추가하여 해당 페이지로 연결될 수 있도록 한다.
- ❑ Home(/sample/home.do), Page Structure(/sample/page.do), List View(/sample/list.do)

```
<ul data-role="listview" data-inset="true">
  <li><a href="/sample/home.do">Home</a></li>
  <li><a href="/sample/page.do">Page Structure</a></li>
  <li><a href="/sample/list.do">List View</a></li>
</ul>
```

Step12. List View

[Step 12-03] List Content Formatting

- List Divider 로 구분된 List View 를 생성한다.
- List View 의 'data-dividertHEME' 속성을 추가하여 List Divider 에 '테마'를 적용하여 빨간색으로 변경한다.
- List Divider 로 구분된 List 의 첫 번째 카테고리의 항목들에 Count Bubble 을 추가한다.
- <h3> Tag 와 <p> Tag 를 이용하여 List View 텍스트에 다양한 효과를 적용한다.

Step12. List View

[Step 12-03] List Content Formatting

```
<ul data-role="listview" data-inset="true" data-dividertHEME="f">
  <li data-role="list-divider">과일</li>
  <li><a>
    <h3>사과</h3>
    <p>무기질이 풍부한 과일</p>
    <span class="ui-li-count">apple</span>
  </a></li>
  <li><a>
    <h3>바나나</h3>
    <p>탄수화물이 풍부한 과일</p>
    <span class="ui-li-count">banana</span>
  </a></li>
  <li data-role="list-divider">야채</li>
  <li><a>
    <h3>오이</h3>
    <p>수분이 많은 야채</p>
  </a></li>
  <li><a>
    <h3>양배추</h3>
    <p>위궤양에 도움이 되는 야채</p>
  </a></li>
  <li><a>
    <h3>당근</h3>
    <p>지용성 비타민이 풍부한 야채</p>
  </a></li>
</ul>
```

Step12. List View

[Step 12-04] Split Button & Search Filter 생성

- ❑ [Step 12-01] 의 List View 에 Split Button 과 Search Filter 를 추가하여 과제를 수행한다.
- ❑ List 항목에 두 개의 <a> Tag 를 추가하여 Split Button 을 생성한다.
- ❑ Split Button 의 링크가 Home("/sample/home.do") 으로 연결되도록 설정한다.
- ❑ List 에 'data-split-icon' 속성을 추가하여 Split Button 의 아이콘을 집 모양(home)으로 변경한다.
- ❑ List 에 'data-filter' 속성을 추가하여 List 상단에 Search Filter 를 생성한다.

```
<ul data-role="listview" data-inset="true" data-split-icon="home" data-filter="true">  
  <li><a href="/sample/home.do">Home</a><a href="/sample/home.do">Home</a></li>  
  <li><a href="/sample/page.do">Page Structure</a><a href="/sample/home.do">Home</a></li>  
  <li><a href="/sample/list.do">List View</a><a href="/sample/home.do">Home</a></li>  
</ul>
```

Step12. List View

[Step 12-05] Nested List 생성

- 하나의 List 를 List 항목(Tag)으로 갖는 Nested List 를 생성한다.
- [Step 12-02] List View 의 List Divider 타이틀을 상위 List 로 갖는 Nested List 를 생성한다.
- 상위 List 의 항목을 클릭하여 List 의 하위 항목 페이지로 이동하는 것을 확인한다.

Step12. List View

[Step 12-05] Nested List 생성

```
<ul data-role="listview" data-inset="true">
  <li>과일
    <ul data-role="listview">
      <li><a>
        <h3>사과</h3>
        <p>무기질이 풍부한 과일</p>
        <span class="ui-li-count">apple</span>
      </a></li>
      <li><a>
        <h3>바나나</h3>
        <p>탄수화물이 풍부한 과일</p>
        <span class="ui-li-count">banana</span>
      </a></li>
    </ul>
  </li>
  <li>야채
    <ul data-role="listview">
      <li><a>
        <h3>오이</h3>
        <p>수분이 많은 야채</p>
      </a></li>
      <li><a>
        <h3>양배추</h3>
        <p>위궤양에 도움이 되는 야채</p>
      </a></li>
      <li><a>
        <h3>당근</h3>
        <p>지용성 비타민이 풍부한 야채</p>
      </a></li>
    </ul>
  </li>
</ul>
```

Step13. Date / Time Picker

[Step 13-01-1] Date / Time Picker 설정

[Step 13-01-2] Date / Time Picker 설정

[Step 13-02] 달력형태 Date Picker 호출

[Step 13-03] 12 시간 단위 Time Picker 호출



Picker

Back

달력형태 Date Picker 호출

날짜 :

12 시간 단위 Time Picker 호출

시간 :

egovFrame

Step13. Date / Time Picker

[Step 13-01-1] Date / Time Picker 설정

- Date / Time Picker 사용을 위하여 JavaScript 코드를 순서대로 추가한다.

```
<script type="text/javascript">
    $( document ).bind( "mobileinit", function(){
        $.mobile.page.prototype.options.degradeInputs.date = 'text';
    });
</script>
```

[Step 13-01-2] Date / Time Picker 설정

- Date / Time Picker 사용을 위하여 JavaScript 코드를 순서대로 추가한다.

```
<script type="text/javascript" src="/js/egovframework/mb/cmm/datepicker/jquery.mobile.datebox.js"></script>
```

Step13. Date / Time Picker

[Step 13-02] 달력형태 Date Picker 호출

- <input> Tag 에 'name', 'id', 'type', 'data-role', 'data-options' 속성을 지정하여 달력형태의 Date Picker 를 호출한다.

```
<input name="mydate" id="defandroid" type="date" data-role="datebox" data-options="{\"mode\": \"calbox\"}" />
```

[Step 13-03] 12 시간 단위 Time Picker 호출

- <input> Tag 에 'name', 'id', 'type', 'data-role', 'data-options' 속성을 지정하여 12 시간 단위의 Time Picker 를 호출한다.

```
<input name="mydate" id="mydate" type="date" data-role="datebox" data-options="{\"mode\": \"timebox\",  
  \"timeFormat\": 12}" />
```

Step14. Check / Radio

[Step 14-01] Check Box 생성

[Step 14-02] Radio Button 생성

[Step 14-03-1] Radio Button 새 로고침

[Step 14-03-2] Radio Button 새 로고침

[Back](#) Check / Radio

Check Box 생성

Check Box :

<input type="checkbox"/>	Check Box 1
<input checked="" type="checkbox"/>	Check Box 2

Radio Button 생성

Radio Button :

<input checked="" type="radio"/>	Radio 1	<input type="radio"/>	Radio 2
----------------------------------	---------	-----------------------	---------

Radio Button 새 로고침

Radio Button :

<input checked="" type="radio"/>	Checked Radio
----------------------------------	---------------

새 로고침

egovFrame

Step14. Check / Radio

[Step 14-01] Check Box 생성

- ❑ `<input>` Tag 의 'type' 속성을 'checkbox' 로 지정하여 Check Box 를 생성한다.
- ❑ 동일 Check Box 에 속해있는 각 항목의 'name' 속성을 동일하게 부여하여 다중선택이 가능하도록 설정한다.
- ❑ `<legend>` Tag 를 추가하여 Check Box 의 타이틀을 명시한다.
- ❑ `<fieldset>` Tag 에 'data-role' 속성을 추가하여 Check Box 그룹을 생성한다.
- ❑ 기본적으로 두 번째 항목에 체크된 효과가 나타나도록 설정하여 두 개의 항목을 갖는 Check Box 를 생성한다.

```
<fieldset data-role="controlgroup">  
  <legend>Check Box : </legend>  
  <input type="checkbox" name="checkbox-1" id="checkbox-1" />  
  <label for="checkbox-1">Check Box 1</label>  
  <input type="checkbox" name="checkbox-2" id="checkbox-2" checked="checked"/>  
  <label for="checkbox-2">Check Box 2</label>  
</fieldset>
```

Step14. Check / Radio

[Step 14-02] Radio Button 생성

- ❑ <input> Tag 의 'type' 속성을 'radio' 로 지정하여 Radio Button 을 생성한다.
- ❑ 동일 Radio Button 에 속해있는 각 항목의 'name' 속성을 유일하게 부여하여 단일선택이 가능하도록 설정한다.
- ❑ <legend> Tag 를 추가하여 Radio Button 의 타이틀을 명시한다.
- ❑ <fieldset> Tag 에 'data-role' 속성을 추가하여 가로로 형성된 Radio Button 그룹을 생성한다.
- ❑ 기본적으로 첫 번째 항목에 체크된 효과가 나타나도록 설정하여 3 개의 항목을 갖는 Radio Button 을 생성한다.

```
<fieldset data-role="controlgroup" data-type="horizontal">
  <legend>Radio Button : </legend>
  <input type="radio" name="radio-1" id="radio-1" value="choice-1" checked="checked" />
  <label for="radio-1">Radio 1</label>
  <input type="radio" name="radio-1" id="radio-2" value="choice-2" />
  <label for="radio-2">Radio 2</label>
</fieldset>
```

Step14. Check / Radio

[Step 14-03-1] Radio Button 새로고침

- Radio Button 을 새로고침할 수 있는 JavaScript 를 추가하여 'id' 속성 값이 'radio' 인 Radio Button 을 새로고침한다.

```
<script type="text/javascript" >  
    function radioRefresh() {  
        $("#radio input[type='radio']").attr("checked",false).checkboxradio("refresh");  
    }  
</script>
```

[Step 14-03-2] Radio Button 새로고침

- '새로고침' 버튼을 클릭하여 Radio Button 이 새로고침 되는 것을 확인한다.

Step15. Icon

[Step 15-01] Icon 생성

[Step 15-02] Icon 위치 변경

[Step 15-03] Icon 형태 및 색상 변경



Step15. Icon

[Step 15-01] Icon 생성

- ❑ 버튼에 'data-icon' 속성을 추가하여 집모양(home)의 Icon 을 생성한다.

```
<a href="/sample/home.do" data-role="button" data-icon="home">Home</a>
```

[Step 15-02] Icon 위치 변경

- ❑ 버튼에 'data-iconpos' 속성을 추가하여 버튼의 위쪽에 Icon 을 위치시킨다.
- ❑ 버튼에 'data-iconpos' 속성을 추가하여 버튼의 아래쪽에 Icon 을 위치시킨다.
- ❑ 버튼에 'data-iconpos' 속성을 추가하여 Icon 만 존재하는 버튼을 생성한다.
- ❑ 별 모양(star) 의 Icon 을 생성한다.

```
<a data-role="button" data-icon="star" data-iconpos="top">위쪽</a>  
<a data-role="button" data-icon="star" data-iconpos="bottom">아래쪽</a>
```

Step15. Icon

[Step 15-03] Icon 형태 및 색상 변경

- ❑ 버튼에 'data-iconpos' 속성을 추가하여 Icon 만 존재하는 버튼을 생성한다.
- ❑ 버튼에 'data-theme' 속성을 변경하여 Icon 의 색상을 변경한다.
- ❑ 버튼에 'data-inline' 속성을 추가하여 Icon 이 나란히 배열될 수 있도록 한다.
- ❑ 돋보기 모양(search) 의 Icon 만 존재하는 버튼을 생성하고 'a'테마를 적용하여 검정색으로 색상을 변경한다.
- ❑ 느낌표 모양(info) 의 Icon 만 존재하는 버튼을 생성하고 'g'테마를 적용하여 초록색으로 색상을 변경한다.
- ❑ 휴대폰 모양(phone) 의 Icon 만 존재하는 버튼을 생성하고 'e'테마를 적용하여 노란색으로 색상을 변경한다.
- ❑ 격자 모양(grid) 의 Icon 만 존재하는 버튼을 생성하고 'f'테마를 적용하여 빨간색으로 색상을 변경한다.

```
<a data-role="button" data-icon="search" data-iconpos="notext" data-theme="a" data-inline="true">돋보기</a>  
<a data-role="button" data-icon="info" data-iconpos="notext" data-theme="g" data-inline="true">느낌표</a>  
<a data-role="button" data-icon="phone" data-iconpos="notext" data-theme="e" data-inline="true">휴대폰</a>  
<a data-role="button" data-icon="grid" data-iconpos="notext" data-theme="f" data-inline="true">격자모양</a>
```

Step16. Selector / Switch

[Step 16-01] Selector 생성

[Step 16-02] Selector 형태변경

[Step 16-03] Switch 생성

Selector 1	<input checked="" type="checkbox"/>
Selector 2	<input checked="" type="checkbox"/>
Selector 3	<input type="checkbox"/>
Selector 4	<input type="checkbox"/>

Back Selector / Switch

Selector 생성

Selector :

Selector 1

Selector 형태변경

Selector :

Switch 생성

Switch :

Off

egovFrame

Step16. Selector / Switch

[Step 16-01] Selector 생성

- ❑ <select> Tag 내부에 네 개의 <option> 을 추가하여 네 가지를 선택할 수 있는 Selector 를 생성한다.
- ❑ <label> Tag 를 추가하여 Selector 의 타이틀을 명시한다.
- ❑ 'fieldcontain' 속성을 갖는 <div> Tag 로 <label> Tag 와 <select> Tag 를 알맞게 배열한다.

```
<div data-role="fieldcontain">
  <label for="select-choice-1" class="select">Selector : </label>
  <select name="select-choice-1" id="select-choice-1">
    <option value="Selector 1">Selector 1</option>
    <option value="Selector 2">Selector 2</option>
    <option value="Selector 3">Selector 3</option>
    <option value="Selector 4">Selector 4</option>
  </select>
</div>
```


Step16. Selector / Switch

[Step 16-02] Selector 형태변경

- ❑ [Step 16-01] 의 Selector 에 'data-native-menu' 속성을 추가하여 Selector 의 형태를 변경한다.
- ❑ Selector 에 'multiple' 속성을 추가하여 다중선택이 가능하도록 설정한다.
- ❑ 'fieldcontain' 속성을 갖는 <div> Tag 로 <label> Tag 와 <select> Tag 를 알맞게 배열한다.

```
<div data-role="fieldcontain">
  <label for="select-choice-2" class="select">Selector :</label>
  <select name="select-choice-2" id="select-choice-2" data-native-menu="false" multiple="multiple">
    <option value="Selector 1">Selector 1</option>
    <option value="Selector 2">Selector 2</option>
    <option value="Selector 3">Selector 3</option>
    <option value="Selector 4">Selector 4</option>
  </select>
</div>
```

Step16. Selector / Switch

[Step 16-03] Switch 생성

- ❑ <select> Tag 의 'data-role' 속성 값을 'slider' 로 지정하여 Switch 를 생성한다.
- ❑ <label> Tag 를 추가하여 타이틀을 명시하고 <option> 값을 'On', 'Off' 로 설정한다.
- ❑ 'fieldcontain' 속성을 갖는 <div> Tag 로 <label> Tag 와 <select> Tag 를 알맞게 배열한다.

```
<div data-role="fieldcontain">
  <label for="slider">Switch :</label>
  <select name="slider" id="slider" data-role="slider">
    <option value="off">Off</option>
    <option value="on">On</option>
  </select>
</div>
```

Step17. Collapsible Block

[Step 17-01] Collapsible Block 생성

[Step 17-02] Collapsible Block 형태변경

Back Collapsible Block

Collapsible Block 생성

default collapsible header

default collapsible header content

open collapsible header

open collapsible header content

Collapsible Block 형태변경

default collapsible header

open collapsible header

open collapsible header content

egovFrame

Step17. Collapsible Block

[Step 17-01] Collapsible Block 생성

- ❑ <collapsible> div Tag 를 2개 생성한다.
- ❑ <H3> Tag 를 추가하여 collapsible 의 타이틀을 명시한다.
- ❑ <p> Tag를 추가하여 collapsible 를 펼쳤을 때 내용을 명시한다.

```
<div data-role="collapsible">  
<h3>default collapsible header</h3>  
<p>default collapsible header content</p>  
</div>  
<div data-role="collapsible" data-collapsed="false">  
<h3>open collapsible header</h3>  
<p>open collapsible header content</p>  
</div>
```

Step17. Collapsible Block

[Step 17-02] Collapsible Block 형태변경

- [Step 17-01] 의 collapsible 에 <data-role="collapsible-set"> Tag를 추가하여 Collapsible를 그룹형태로 변경한다.

```
<div data-role="collapsible-set">
<div data-role="collapsible">
<h3>default collapsible header</h3>
<p>default collapsible header content</p>
</div>
<div data-role="collapsible" data-collapsed="false">
<h3>open collapsible header</h3>
<p>open collapsible header content</p>
</div>
</div>
```