

jQuery Mobile Basic

jQuery Mobile 1.0 Beta 2

문서 자체 버전 : 0.2

원문 : jquerymobile.com/demos/1.0b2/

작성자 : 황현재(닉네임:모지씨)

이메일 : hhjae84@gmail.com

카페 : <http://cafe.naver.com/dflashcafe>

※ 이 문서는 개인적으로 작성한 내용으로 원문 대비 추가/삭제 및 요약된 부분이 상당수 있으며, 작성자 주관에 의하여 작성된 내용이 포함되어있습니다. jQuery Mobile 의 버전업에 의해 내용이 달라질 수 있습니다. 내용상의 문제점은 이메일이나 카페에 글 남겨주시면 반영하도록 하겠습니다. 감사합니다.

목차

1장: 소개

2장: 컴포넌트

페이지 및 대화창(Pages & dialogs)

- 페이지 구조(Anatomy of a page)	7
- 단일 페이지(Single page template)	9
- 다중 페이지(Multi-page template)	10
- 페이지 제목(Page title)	11
- 페이지 링크(Linking pages)	12
- 페이지 전환(Page transitions)	13
- 대화창(Dialogs)	14
- 페이지 로드(Preload & cache pages)	16
- 페이지관련 스크립트(Scripting pages)	17
- 페이지 테마(Theming pages)	19

툴바(Toolbars)

- 툴바 기본(Toolbar basics)	20
- 상단 바(Header bars)	21
- 하단 바(Footer bars)	24
- 메뉴 바(Navbars)	26
- 위치 고정(Fixed positioning)	28
- 전체화면에서 위치(Fullscreen positioning)	28
- 따라다니는 하단 메뉴 바(Persistent footer nav bar)	29

버튼(Buttons)

- 버튼 마크업 옵션(Button markup options)	30
- 버튼 아이콘(Button icons)	31
- 일렬로 된 버튼(Inline buttons)	36
- 버튼 그룹(Grouped buttons)	37
- 버튼 테마(Theming buttons)	38

컨텐츠 서식(Content formatting)

- 그리드 레이아웃(Layout grids(columns))	39
- 내용을 접을 수 있는 블록(Collapsible content blocks)	42
- 내용을 접을 수 있는 블록 세트(아코디언)(Collapsible sets(accordions))	44
- 내용 테마(Theming contents)	46

폼 요소(Form elements)

- 기본 폼(Form basics)	48
- 텍스트 입력(Text inputs)	51
- 검색 입력(Search inputs)	55
- 슬라이더(Slider)	56
- 토글 스위치(Flip toggle switch)	57
- 라디오 버튼(Radio buttons)	59
- 체크박스(Checkboxes)	63
- 선택 메뉴(Select menus)	67
- 폼 테마(Theming forms)	81
- 관련 메서드(Plugin methods)	82

리스트(List views)

- 읽기 전용 리스트(Read-only list)	85
- 기본 링크 리스트(Basic Inked list)	86
- 중첩 리스트(Nested list)	87
- 번호가 붙은 리스트(Numbered list)	88

- 버튼이 나뉜 리스트(Split button list)	89
- 리스트 구분자(List dividers)	90
- 카운트 풍선(Count bubble)	92
- 썸네일 리스트(Thumbnails)	93
- 아이콘 리스트(Icons)	94
- 내용 설정(Content formatting)	95
- 리스트 검색 바(Search filter bar)	98
- 안으로 들어간 리스트 검색 바(Inset search filter bar)	101
- 구분자가 있는 리스트 검색 바(Search filter bar with dividers)	102
- 리스트 테마(Theming lists)	104
- 관련 메서드(Plugin methods)	109

3장: API

기본값 설정(Configuring defaults)

- jQuery Mobile 초기화 이벤트(The mobileinit event)	101
- 구성 옵션(Configurable options)	111

이벤트(Events)

- 터치 이벤트(Touch events)	115
- 방향 전환 이벤트(Orientation change event)	117
- 스크롤 이벤트(Scroll events)	118
- 페이지 표시 관련 이벤트(Page show/hide events)	119
- 페이지 초기화 이벤트(Page initialization events)	122
- 가상 마우스 이벤트(Virtual mouse events)	123
- 애니메이션 이벤트(Animation events)	123

메서드 & 유틸리티(Methods & utilities)

- 화면 변경(changePage)	124
- 페이지 로드(loadPage)	125
- 로딩메세지 보여주기(showPageLoadingMsg)	126
- 로딩메세지 숨기기(hidePageLoadingMsg)	126
- URL 정보(parseUrl)	127
- 절대경로 생성(makePathAbsolute)	127
- 도메인 내에서의 절대경로 생성(makeUrlAbsolute)	128
- 같은 도메인 여부 확인(isSameDomain)	129
- 상대경로 여부 확인(isRelativeUrl)	130
- 절대경로 여부 확인(isAbsoluteUrl)	131
- 스크롤 이동(silentScroll)	132
- 해상도 최대 최소값 설정(addResolutionBreakpoints)	132
- 활성화된 페이지 찾기(activePage)	133

반응하는 레이아웃(Responsive Layout)

- 방향에 따른 클래스(Orientation Classes)	134
- 크기에 따른 클래스(Min/Max Width Breakpoint Classes)	135
- 크기 제어 포인트(Adding Width Breakpoints)	136
- 미디어 쿼리(Running Media Queries)	136

1장: 소개

jQuery Mobile 은 스마트폰과 태블릿을 위한 터치에 최적화 된 웹 프레임워크로 다음과 같은 목표 및 전략을 가지고 있습니다.

- 플랫폼과 디바이스에 영향을 받지 않는 프레임워크
 - iOS, ANDROID, BlackBerry, bada, Windows Phone, palm webOS, Symbian, MeeGo
- 터치에 최적화된 레이아웃과 UI 위젯
- 쉬운 모바일 테마 구축

그리고 아래와 같은 스폰서들의 지원을 받고 있습니다.

filament group, mozilla corporation, palm, BlackBerry, NOKIA Connectiong People, DeviceAtlas, Adobe, dotMobi, ORIGINATE

2장: 컴포넌트

페이지 및 대화창(Pages & dialogs)

페이지 구조(Anatomy of a page)

jQuery Mobile 구조

jQuery Mobile 사이트는 HTML5 Doctype 을 사용합니다.

그리고 jQuery, jQuery Mobile, jQuery Mobile theme CSS 를 요구합니다.

```
<!DOCTYPE html>
<html>
  <head>
    <title>jQuery Mobile page</title>
    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.0b2/jquery.mobile-1.0b2.min.css"
  />
    <script src="http://code.jquery.com/jquery-1.6.2.min.js"></script>
    <script src="http://code.jquery.com/mobile/1.0b2/jquery.mobile-1.0b2.min.js"></script>
  </head>
  <body>
    ...
  </body>
</html>
```

뷰포트 메타태그

메타태그를 이용해 뷰포트 속성을 조절하여 장치 화면의 너비를 설정할 수 있습니다. 너비의 단위는 픽셀 크기로 설정할 수 있습니다.(아이폰, 안드로이드폰의 경우 320px 인 것으로 보입니다.)

```

<!DOCTYPE html>
<html>
  <head>
    <title>jQuery Mobile page</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.0b2/jquery.mobile-1.0b2.min.css" />
    <script src="http://code.jquery.com/jquery-1.6.2.min.js"></script>
    <script src="http://code.jquery.com/mobile/1.0b2/jquery.mobile-1.0b2.min.js"></script>
  </head>
  <body>
    ...
  </body>
</html>

```

페이지 요소

<body> 태그 내부에 페이지를 만들어 사용합니다.

페이지를 만드는 요소는 div 를 사용하며 data-role="page" 속성을 추가해주면 됩니다.

```

<div data-role="page">
  ...
</div>

```

전형적인 페이지의 요소로 상단 바와 내용, 하단바가 들어갑니다.

이 요소들은 페이지의 자식요소의 div 에 data-role 속성을 "header"와 "content", "footer" 로 설정해주는 것으로 가능합니다.

```

<div data-role="page">
  <div data-role="header">...</div>
  <div data-role="content">...</div>
  <div data-role="footer">...</div>
</div>

```


단일 페이지(Single page template)

단일 페이지

```
<!DOCTYPE html>
<html>
  <head>
    <title>jQuery Mobile page</title>
    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.0b2/jquery.mobile-1.0b2.min.css />
    <script src="http://code.jquery.com/jquery-1.6.2.min.js"> </script>
    <script src="http://code.jquery.com/mobile/1.0b2/jquery.mobile-1.0b2.min.js"> </script>
  </head>
  <body>
    <!-- 페이지 -->
    <div data-role="page">
      <!-- 상단 바 -->
      <div data-role="header">
        <h1>Page title</h1>
      </div>
      <!-- 내용 -->
      <div data-role="content">
        <p>Page contents</p>
      </div>
      <!-- 하단 바 -->
      <div data-role="footer">
        <h1>Page Footer</h1>
      </div>
    </div>
  </body>
</html>
```

다중 페이지(Multi-page template)

다중 페이지

파일 내부에 여러 페이지를 만들 수 있습니다.

다중 페이지에서의 이동은 페이지의 id 로 이동 가능합니다.

```
<body>
  <!-- 첫 페이지 -->
  <div data-role="page" id="first">
    <div data-role="header">
      <h1>First page</h1>
    </div>
    <div data-role="content">
      <p>Page contents</p>
      <p><a href="#second">Go to second page</a> </p>
    </div>
    <div data-role="footer">
      <h1>First page Footer</h1>
    </div>
  </div>
  <div data-role="page" id="second">
    <div data-role="header">
      <h1>Second page title</h1>
    </div>
    <div data-role="content">
      <p>Page contents</p>
      <p><a href="#first">Go to first page</a> </p>
    </div>
    <div data-role="footer">
      <h1>Second page Footer</h1>
    </div>
  </div>
</body>
```

페이지 제목(Page title)

페이지 제목

페이지 이동시 ajax 방식을 기본으로 사용하기 때문에 head 내부에 title 로 작성된 페이지 제목은 변경되지 않습니다. 페이지 제목을 변경하기 위해서는 data-title 속성을 이용하면 자동으로 업데이트 됩니다.

```
<div data-role="page" data-title="page title">  
    ...  
</div>
```

페이지 링크(Linking pages)

페이지 링크

기본적으로 새로운 페이지는 ajax 를 이용해 페이지를 보여줍니다.

이 방법을 사용하고 싶지 않다면 `rel="external"` 또는 `data-ajax="false"` 속성을 추가해주면 됩니다.

```
<a href="newpage.html" rel="external">New page link</a>
```

뒤로가기 링크

링크로 이전페이지로 돌아가기 위해서는 `data-rel="back"` 속성을 추가하면 됩니다.

만약 페이지 전환이 페이지 내부에 들어왔던 반대로 동작하게 하려면 `data-direction="reverse"` 속성을 추가해주어야 합니다.

```
<a href="newpage.html" data-rel="back">New page link</a>
<a href="newpage.html" data-rel="back" data-direction="reverse">New page link</a>
```

페이지 전환(Page transitions)

페이지 전환

기본적으로 6개의 페이지 전환 효과를 제공합니다.
data-transition 속성을 추가해 사용할 수 있습니다.

기본적인 전환 효과

- slide
- slideup
- slidedown
- pop
- fade
- flip

```
<a href="newpage.html" data-transition="pop">New page link</a>
```

대화창(Dialogs)

대화창 생성

링크 속성에 `data-rel="dialog"` 를 추가하면 대화창 형식으로 페이지가 표시됩니다.

```
<a href="newpage.html" data-rel="dialog">New page link</a>
```

전환 효과

대화창의 내장된 전환 효과는 `pop` 입니다.

대화창과 같은 느낌을 위해 아래 효과 중 하나를 지정하는 것이 좋습니다.

- `slidedown`
- `pop`
- `flip`

대화창 닫기

대화창의 경우 자동으로 왼쪽 상단에 닫기버튼이 표시됩니다.

임의로 닫기 버튼을 만들기 위해서는 `a` 태그의 `href` 속성에 `"#"`을 설정해주고 `data-rel="back"` 를 추가해주면 됩니다.

자바스크립트를 이용한 대화창 닫기

```
$(".ui-dialog").dialog("close");
```

뒤로가기 버튼으로 대화창이 오픈되기 전 상태로 돌아가기

```
<a href="#" data-rel="back">New page link</a>
```

닫기 버튼의 글자는 mobileinit 이벤트를 설정해주고 변경해야 합니다.

```
$.mobile.dialog.prototype.options.closeBtnText = "Close";
```

페이지 로드(Preload & cache pages)

페이지 로드

페이지를 백그라운드에서 로드합니다. 로딩메세지는 표시되지 않습니다.

```
<a href="page.html" data-prefetch>link</a>
```

DOM 크기 관리

페이지 전환이 있는 페이지들은 모두 하나의 DOM 에 있습니다. 하지만 이 방법은 페이지에 새로 고침이 일어나기 전에 계속 남아있기 때문에 일부 장치에서 메모리 관리에 문제점이나 속도 저하 및 충돌을 일으킬 우려가 있습니다.

domCache 설정을 이용해 DOM 을 깨끗하게 유지할 수 있습니다. 기본적으로 false 값을 가집니다.

기본값 변경

```
$.mobile.page.prototype.options.domCache = "true";
```

링크별 설정

```
<a href="page.html" data-dom-cache="true">link</a>
```

프로그래밍 방식 설정

```
elem.page({domCache: true});
```


페이지 관련 스크립트(Scripting pages)

페이지 생성

페이지가 생성되었다는 이벤트는 DOM 준비된 상태를 알려줍니다. 페이지 관련 스크립트는 pagecreate 이벤트 이후 사용하는 것이 좋습니다.

```
$("#page").live("pagecreate", function(e)
{
    alert("DOM READY");
});
```

페이지 변경

스크립트로 페이지를 변경해야 하는 경우 change() 메서드를 사용할 수 있습니다.

slideup 방식으로 페이지 전환

```
$.mobile.changePage("page.html", {transition: "slideup"});
```

데이터를 포함한 페이지 전환

```
$.mobile.changePage("complete.php ", {type:"post", data:$("form").serialize()});
```

페이지 로드

외부 페이지를 불러와 DOM 에 삽입합니다.

```
$.mobile.loadPage("page.html");
```

페이지 내 위치 이동

페이지의 스크롤 위치를 이동시킵니다.

```
$.mobile.silentScroll(300);
```

마우스 및 터치 이벤트

클릭 이벤트 같은 경우 일반적으로 0.5~0.7 초 정도의 시간이 지연됩니다. 이러한 지연을 방지하기 위해 터치 이벤트를 사용할 수 있지만, 일부 모바일 플랫폼(WP7, 블랙베리)에서는 터치 이벤트를 지원하지 않습니다. 이 문제를 해결하기 위해 두가지 유형의 이벤트를 같이 사용하면 이벤트가 중복으로 발생하게 됩니다.

jQuery Mobile 에서는 다음과 같은 가상 이벤트를 이용해 해결합니다. vmouseover, vmousedown, vmousemove, vmouseup, vclick, vmousecancel

페이지 테마(Theming pages)

페이지 테마

페이지 테마는 page 로 설정된 div 태그에 data-theme 속성을 지정하면 변경됩니다.

data-theme 를 지정하지 않을 경우 기본 테마로 설정되며, 기본 테마 이외에 a~e 까지 5개의 테마가 더 있습니다.

```
<div data-role="page" data-theme="a">  
    ...  
</div>
```

툴바(Toolbars)

툴바 기본(Toolbar basics)

툴바 유형

jQuery Mobile 표준으로 header 와 footer 가 있습니다.

header 는 일반적으로 페이지의 첫번째 요소로 제목 역할을 하며 제목 및 최대 두 개의 버튼을 포함되어 있습니다. footer 는 일반적으로 페이지의 마지막 요소이며 내용 및 기능적인 면에서 header 보다 더 자유롭습니다.

툴바 위치 옵션

툴바는 일반적으로 페이지의 상/하단에 자연스럽게 배치되지만 몇 가지 다른 방법으로 페이지에 위치시킬 수 있습니다.

툴바를 고정시켜놓을 수 있으며, 전체화면의 상/하단에 위치시킬 수도 있습니다.

위치의 고정은 data-position="fixed" 로 설정하면 header 는 브라우저의 상단에 footer 는 브라우저의 하단에 위치하게 됩니다.

data-fullscreen="true" 으로 설정하면 페이지를 클릭하는 경우에만 보여지고 숨겨지게 되며 투바 뒤에 내용은 투명하게 보여집니다.

```
<div data-role="header" data-position="fixed">...</div>
```

```
<div data-role="header" data-fullscreen="true">...</div>
```

상단 바(Header bars)

상단바 구조

header 는 페이지 상단의 톨바 입니다.

제목은 일반적으로 h1 태그를 이용해 작성하지만 페이지의 유연성을 위해 h1~h6 어떤 태그로도 가능합니다. 하지만 모든 스타일은 동일합니다.

a 테마가 기본으로 설정되어 있습니다.

```
<div data-role="header">
  <h1>Page title</h1>
</div>
```



버튼 추가

버튼은 a 태그로 추가할 수 있습니다. 기본적인 위치는 첫번째 요소를 왼쪽에 배치하고 두번째 요소를 오른쪽에 배치합니다.

```
<div data-role="header">
  <a href="index.html">Back</a>
  <h1>Page title</h1>
  <a href="index.html">Save</a>
</div>
```



버튼은 기본적으로 header 의 테마를 상속받습니다. 다른 테마로 설정하기 위해서는 a 태그에 테마를 지정해 주어야 합니다.

```
<div data-role="header">
  <a href="index.html" data-theme="b">Back</a>
  <h1>Page title</h1>
  <a href="index.html" data-theme="c">Save</a>
</div>
```



기본적으로 버튼을 하나만 추가하는 경우 왼쪽에 표시되지만 클래스를 지정해 오른쪽으로 위치시킬 수 있습니다. ui-btn-right 클래스를 지정해주면 버튼이 오른쪽에 위치하게 됩니다.

```
<div data-role="header">
  <a href="index.html" class="ui-btn-right">Home</a>
  <h1>Page title</h1>
</div>
```



뒤로가기 버튼 추가

기본적으로 뒤로가기 버튼은 없지만 jQuery Mobile 에는 뒤로가기 버튼을 자동으로 추가해주는 기능이 있습니다. data-add-back-btn="true" 로 설정하면 뒤로가기 버튼이 생성됩니다. (뒤로 갈 페이지가 없는 경우 생성되지 않습니다.) a 태그를 생성해서 data-rel="back" 으로 설정해도 됩니다.

```
<div data-role="header" data-add-back-btn="true">
  <h1>Page title</h1>
```

```
</div>
```



```
<div data-role="header" >
  <a href="#" data-rel="back">Back</a>
  <h1>Page title</h1>
</div>
```



뒤로가기 버튼 텍스트 변경

data-add-back-btn 설정으로 뒤로가기 버튼의 글자를 변경할 수 있습니다.

```
$.mobile.page.prototype.options.backBtnText = "previous";
```

뒤로가기 버튼 기본 테마 변경

data-add-back-btn 설정으로 생성된 뒤로가기 버튼의 테마를 변경할 수 있습니다.

```
$.mobile.page.prototype.options.backBtnTheme = "a";
```

하단 바(Footer bars)

하단바 구조

하단바는 바닥글의 역할을 합니다. 기본적으로 a 테마가 적용되어 있습니다.

```
<div data-role="footer">
  <h1>Page footer</h1>
</div>
```



버튼 추가

footer 에는 다른 요소를 수용할 수 있는 패딩이 없습니다. 패딩을 포함하려면 "ui-bar" 클래스를 적용해야 합니다.

```
<div data-role="footer" class="ui-bar">
  <a href="index.html" data-role="button">Add</a>
</div>
```



고정되어있는 하단바

페이지 전환시 하단바가 이동되지만 고정되어 있게 할 수 있습니다. 전환되는 동안 footer 를 유지하려면 페이지와 전환되는 페이지 내 footer 의 data-id 가 동일하게 설정되어야 합니다.

효과가 제대로 동작하기 위해서는 data-position="fixed" 로 설정할 필요가 있습니다.

```
<body>
  <!-- 첫 페이지 -->
  <div data-role="page" id="first">
    <div data-role="header">
      <h1>First page</h1>
    </div>
    <div data-role="content">
      <p>Page contents</p>
      <p><a href="#second">Go to second page</a> </p>
    </div>
    <div data-role="footer" data-id="fixedFooter">
      <h1>Page Footer</h1>
    </div>
  </div>
  <div data-role="page" id="second">
    <div data-role="header">
      <h1>Second page title</h1>
    </div>
    <div data-role="content">
      <p>Page contents</p>
      <p><a href="#first">Go to first page</a> </p>
    </div>
    <div data-role="footer" data-position="fixed" data-id="fixedFooter">
      <h1>Page Footer</h1>
    </div>
  </div>
</body>
```

메뉴 바(Navbars)

간단한 메뉴바

메뉴바의 버튼은 5개까지는 한 줄로 보이고 6개부터는 두 칸으로 나누어 여러 줄로 보여집니다. 메뉴바 사용을 위해서는 컨테이너 요소에 `data-role="navbar"` 속성을 적용하여 리스트로 작성해야 화면 크기에 맞게 보여집니다.

메뉴바 내의 버튼의 활성화를 위해서는 `"ui-btn-active"` 클래스를 적용해주면 됩니다.

```
<div data-role="footer" >
  <div data-role="navbar">
    <ul>
      <li><a href="#" class="ui-btn-active">One</a> </li>
      <li><a href="#">Two</a> </li>
    </ul>
  </div>
</div>
```

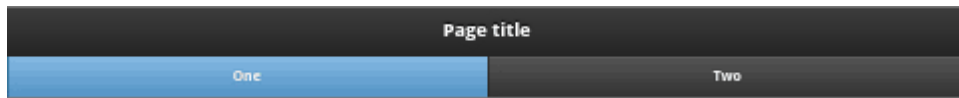


상단바 안의 메뉴바

header 에 메뉴바를 넣는 경우 제목이 나오는 라인 바로 아래에 메뉴바가 보여집니다.

```
<div data-role="header" >
  <h1>Page title</h1>
  <div data-role="navbar">
    <ul>
      <li><a href="#" class="ui-btn-active">One</a> </li>
      <li><a href="#">Two</a> </li>
    </ul>
  </div>
</div>
```

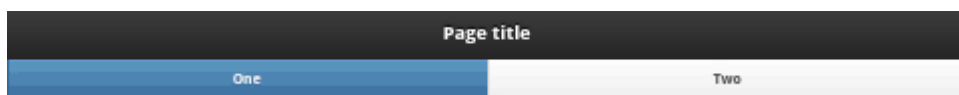
```
</div>
```



메뉴바 테마

메뉴바의 버튼에 테마를 적용하기 위해서는 각각의 요소에 data-theme 를 적용해주어야 합니다.

```
<div data-role="header" >
  <h1>Page title</h1>
  <div data-role="navbar">
    <ul>
      <li><a href="#" data-theme="b">One</a></li>
      <li><a href="#" data-theme="c">Two</a></li>
    </ul>
  </div>
</div>
```



위치 고정(Fixed positioning)

위치 고정

툴바의 위치를 고정하기 위해서는 `data-position="fixed"` 속성을 설정해주어야 합니다. 스크롤 중에는 잠시 숨겨지며, 스크롤이 멈추면 다시 보이게 됩니다.

```
<div data-role="header" data-position="fixed">
  <h1>title</h1>
</div>
```

전체화면에서 위치(Fullscreen positioning)

전체화면에서 위치

툴바가 화면 상/하단에 보여지는 것은 고정하는 것과 같지만 화면 터치시 투바가 숨겨지고 보여지는 것이 반복되며, 투바 뒤에 내용이 들어갈 수 있게 되어 뒷부분의 내용은 투명하게 표시됩니다.

```
<div data-role="footer" data-fullscreen="true">
  <h1>title</h1>
</div>
```

따라다니는 하단 메뉴 바(Persistent footer nav bar)

고정된 하단 메뉴 바

하단 메뉴 바를 페이지가 이동할 때도 고정해두기 위해서는 data-id 속성을 전환되는 페이지와 전환되기 전 페이지를 같게 설정해주면 페이지 전환될 때도 하단 메뉴바는 움직이지 않습니다.

```
<body>
  <!-- 첫 페이지 -->
  <div data-role="page" id="first">
    <div data-role="header">
      <h1>First page</h1>
    </div>
    <div data-role="content">
      <p>Page contents</p>
      <p><a href="#second">Go to second page</a> </p>
    </div>
    <div data-role="footer" data-id="fixedFooter">
      <h1>Page Footer</h1>
    </div>
  </div>
  <div data-role="page" id="second">
    <div data-role="header">
      <h1>Second page title</h1>
    </div>
    <div data-role="content">
      <p>Page contents</p>
      <p><a href="#first">Go to first page</a> </p>
    </div>
    <div data-role="footer" data-position="fixed" data-id="fixedFooter">
      <h1>Page Footer</h1>
    </div>
  </div>
</body>
```

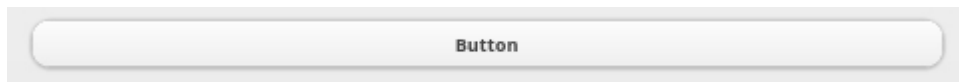
버튼(Buttons)

버튼 마크업 옵션(Button markup options)

링크를 버튼 스타일로 변경

링크에 `data-role="button"` 을 추가하여 버튼 스타일로 변경해줄 수 있습니다.

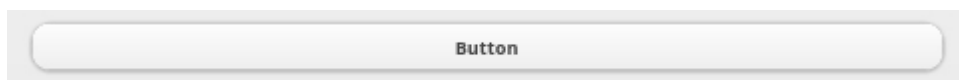
```
<a href="#" data-role="button">Button</a>
```



폼 버튼

input 요소 중 `button`, `submit`, `reset`, `image` 요소도 버튼 스타일로 보여집니다.

```
<input type="button" value="Button" />  
<input type="submit" value="Button" />  
<input type="reset" value="Button" />  
<input type="image" value="Button" />
```



버튼 아이콘(Button icons)

버튼 아이콘

jQuery Mobile 의 버튼 아이콘은 다운로드 크기를 최소화 하기 위해 흰색의 단일 아이콘에 반투명 검은색 서클로 구성되어 있습니다.

아이콘 추가는 data-icon="delete" 와 같이 속성을 추가해주어야 합니다.

```
<a href="#" data-role="button" data-icon="delete">delete</a>
```

기본으로 제공되는 아이콘

- Left arrow - data-icon="arrow-l"



- Right arrow - data-icon="arrow-r"



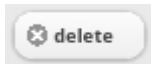
- Up arrow - data-icon="arrow-u"



- Down arrow - data-icon="arrow-d"



- Delete - data-icon="delete"



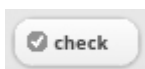
- Plus - data-icon="plus"



- Minus - data-icon="minus"



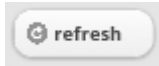
- Check - data-icon="check"



- Gear - data-icon="gear"



- Refresh - data-icon="refresh"



- Forward - data-icon="forward"



- Back - data-icon="back"



- Grid - data-icon="grid"



- Star - data-icon="star"



- Alert - data-icon="alert"



- Info - data-icon="info"



- Home - data-icon="home"



- Search - data-icon="search"



버튼에서의 아이콘 위치

기본적으로 아이콘의 위치는 왼쪽에 배치됩니다. 이 위치는 `data-iconpos` 속성을 통해 변경할 수 있습니다.

```
<!-- 기본(왼쪽) -->
<a href="#" data-role="button" data-icon="arrow-l">default</a>
<!-- 오른쪽 -->
<a href="#" data-role="button" data-icon="arrow-r" data-iconpos="right">right</a>
<!-- 위 -->
<a href="#" data-role="button" data-icon="arrow-up" data-iconpos="top">top</a>
<!-- 아래 -->
<a href="#" data-role="button" data-icon="arrow-down" data-iconpos="bottom">bottom</a>
```



아이콘만 있는 버튼

아이콘이 설정된 버튼에 `data-iconpos="notext"` 로 설정해주면 버튼의 텍스트는 보이지 않고 아이콘만 보여지게 됩니다.

```
<a href="#" data-role="button" data-icon="home" data-iconpos="notext">home</a>
```



사용자 아이콘

만든 아이콘을 적용하려면 CSS 클래스를 만들어주어야 합니다.

클래스명은 `ui-icon-myicon` 과 같이 `ui-icon` 을 붙여서 작성해주어야 합니다. 이미지 크기는 `18x18px` 을 PNG-8 형식을 권장합니다.

data-icon 설정을 통한 사용자 아이콘

```
<style>
    .ui-icon-myicon {
        background-image: url(icon/myicon.png);
        background-repeat: no-repeat;
    }
</style>
```

```
<a href="#" data-role="button" data-icon="myicon">myicon</a>
```



id 설정을 통한 사용자 아이콘

```
<style>
    #myicon .ui-icon {
        background-image: url(icon/myicon.png);
        background-repeat: no-repeat;
    }
</style>
```

```
<a href="#" data-role="button" data-icon="custom" id="myicon">myicon</a>
```

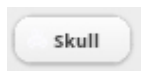


반투명한 배경 숨기기

```
<style>
```

```
#myicon .ui-icon {  
    background-image: url(icon/myicon.png);  
    background-repeat: no-repeat;  
    background-color: transparent;  
}  
</style>
```

```
<a href="#" data-role="button" data-icon="custom" id="myicon">myicon</a>
```



(아이콘이 흰색으로 만들어져 잘 보이지 않습니다.)

아이콘 테마

아이콘의 테마는 아이콘이 이미지 파일로 되어있기 때문에 달라지는 것이 없습니다. 하지만 아이콘 모양은 흰색으로 주위에는 검은색 반투명한 원이 있는 이미지이기 때문에 테마 별로 달라지는 것처럼 느껴집니다.



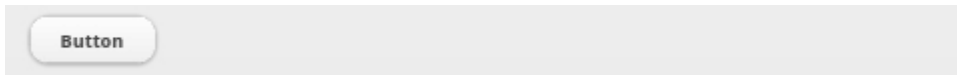
일렬로 된 버튼(Inline buttons)

일렬로 된 버튼

기본적으로 버튼이 적용되면 블록 요소의 스타일을 가집니다.

만약 인라인 요소를 원하는 경우 `data-inline="true"` 속성을 설정해주어야 합니다.

```
<a href="#" data-role="button" data-inline="true">Button</a>
```



여러 버튼이 있는 경우 컨테이너에 적용해주어도 됩니다.

(jQuery Mobile 사이트에는 된다고 설명되어 있는데 실제로는 버그인지 적용되지 않습니다.)

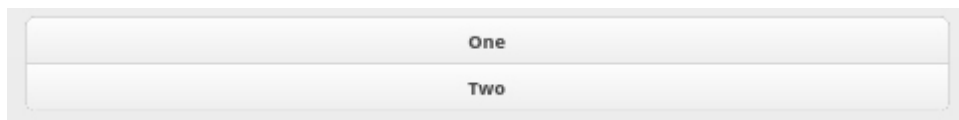
```
<div data-inline="true">
  <a href="#" data-role="button" >One</a>
  <a href="#" data-role="button" >Two</a>
</div>
```

버튼 그룹(Grouped buttons)

버튼 그룹

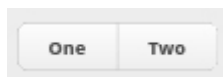
버튼을 시각적으로 하나의 블록으로 형성해줄 수 있습니다. 버튼을 그룹 형식으로 만들어주기 위해서는 컨테이너에 `data-role="controlgroup"` 속성을 지정해주어야 합니다.

```
<div data-role="controlgroup">  
    <a href="#" data-role="button" >One</a>  
    <a href="#" data-role="button" >Two</a>  
</div>
```



기본적으로 버튼이 세로로 배치되지만 가로로 배치되게 설정하려면 `data-type="horizontal"` 을 적용해주어야 합니다.

```
<div data-role="controlgroup" data-type="horizontal">  
    <a href="#" data-role="button">One</a>  
    <a href="#" data-role="button">Two</a>  
</div>
```



버튼 테마(Theming buttons)

버튼 테마

버튼에 다른 테마를 적용하기 위해서는 적용하고자 하는 버튼에 `data-theme` 속성을 설정해주면 됩니다.

```
<a href="#" data-role="button" data-theme="b">Two</a>
```



컨텐츠 서식(Content formatting)

그리드 레이아웃(Layout grids(columns))

그리드 레이아웃

여러 열의 레이아웃을 사용하는 것은 일반적으로 좁은 화면 너비의 모바일 장치에서 권장하지만 작은 요소를 나란히 배치해야 할 경우도 있습니다. jQuery Mobile 에서는 블록 스타일의 클래스를 CSS 기반의 열을 구축할 수 있는 방법을 제공하기 위해 "ui-grid" 클래스를 사용합니다.

행은 ui-grid, 열은 ui-block 를 이용해 표현할 수 있습니다.
추가되는 만큼 행과 열은 증가합니다.

두 열 그리드

두 열의 레이아웃은 각 열이 50% 크기로 배치됩니다.

열이 하나씩 늘어날 때마다 "ui-block" 뒤의 알파벳을 알파벳 순으로 증가시킵니다.

```
<div class="ui-grid-a">
  <div class="ui-block-a">첫 번째 열</div>
  <div class="ui-block-b">두 번째 열</div>
</div>
```

첫 번째 열

두 번째 열

삼 열 그리드

삼 열의 레이아웃은 각 열이 1/3 크기를 가집니다.

두 열 그리드일 경우 ui-grid-a 를 설정하였는데 이 클래스도 열이 증가할 때마다 ui-grid 뒤의 알파벳도 증가시켜주어야 합니다.

두 열일 경우 ui-grid-a, 삼 열일 경우 ui-grid-b, 사 열일 경우 ui-grid-c 와 같은 식입니다.

```
<div class="ui-grid-b">
  <div class="ui-block-a">첫 번째 열</div>
  <div class="ui-block-b">두 번째 열</div>
  <div class="ui-block-c">세 번째 열</div>
</div>
```

첫 번째 열

두 번째 열

세 번째 열

다중 행 그리드

삼 열 그리드를 삼 행을 추가해 3x2 의 그리드를 만드는 내용입니다.

삼 열이니 ui-grid-b 가 적용되었고 열의 내용들도 a 부터 c 까지 증가를 반복하였습니다.

```
<div class="ui-grid-b">
  <div class="ui-block-a">첫 번째 행, 첫 번째 열</div>
  <div class="ui-block-b">첫 번째 행, 두 번째 열</div>
  <div class="ui-block-c">첫 번째 행, 세 번째 열</div>
  <div class="ui-block-a">두 번째 행, 첫 번째 열</div>
  <div class="ui-block-b">두 번째 행, 두 번째 열</div>
  <div class="ui-block-c">두 번째 행, 세 번째 열</div>
</div>
```

첫 번째 행, 첫 번째 열
두 번째 행, 첫 번째 열
세 번째 행, 첫 번째 열

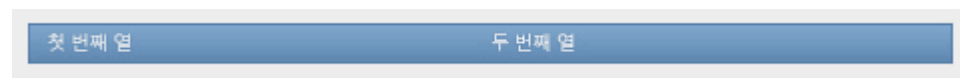
첫 번째 행, 두 번째 열
두 번째 행, 두 번째 열
세 번째 행, 두 번째 열

첫 번째 행, 세 번째 열
두 번째 행, 세 번째 열
세 번째 행, 세 번째 열

그리드의 테마 적용

그리드에는 data-theme 속성이 존재하지 않습니다. 기본적인 패딩을 적용하는 경우 "ui-bar" 클래스를 적용해주어야 하고 테마를 적용하는 경우 "ui-bar-테마명" 을 적용해주어야 합니다.

```
<div class="ui-grid-a ui-bar ui-bar-b">
  <div class="ui-block-a">첫 번째 열</div>
  <div class="ui-block-b">두 번째 열</div>
</div>
```



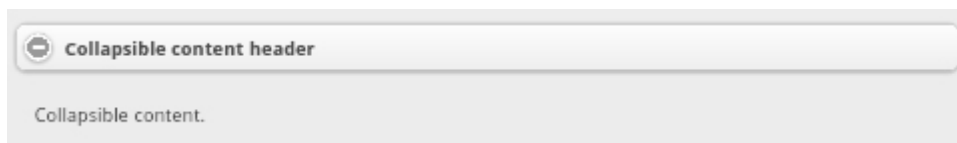
내용을 접을 수 있는 블록(Collapsible content blocks)

내용을 접을 수 있는 블록

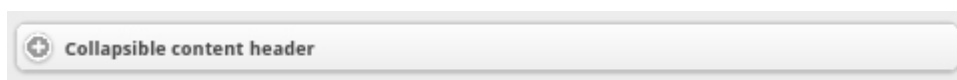
내용을 접을 수 있게 만들려면 컨테이너를 생성하고 `data-role="collapsible"` 속성을 지정해야 합니다. 컨테이너 내부에 제목을 `h1~h6` 태그로 추가합니다. 제목이 추가되면 제목 옆에 `+` 아이콘이 생겨나며 터치할 경우 내용이 펼쳐지면서 `-` 아이콘으로 변경됩니다.

```
<div data-role="collapsible">
  <h3>Collapsible content header</h3>
  <p>Collapsible content.</p>
</div>
```

펼쳐진 화면

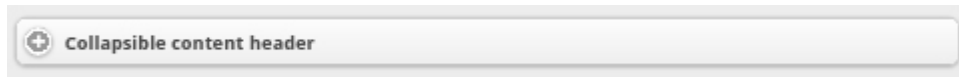


접힌 화면



기본적으로 내용이 펼쳐져 있지만 접혀있는 상태에서 시작하기 위해서는 `data-collapsed="true"` 속성을 지정해야 합니다.

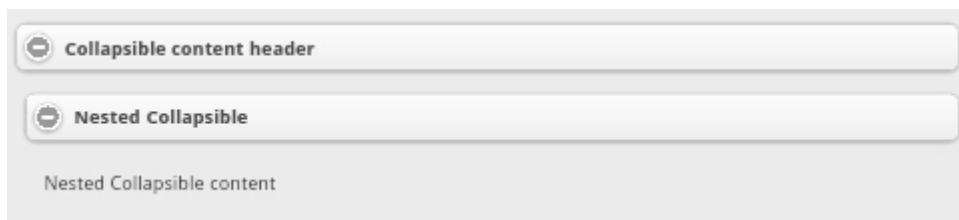
```
<div data-role="collapsible" data-collapsed="true">
  <h3>Collapsible content header</h3>
  <p>Collapsible content.</p>
</div>
```



중첩된 내용을 접을 수 있는 블록

내용을 접을 수 있는 블록의 제목 이외의 부분에 내용을 접을 수 있는 블록을 추가하면 중첩된 블록을 만들 수 있습니다.

```
<div data-role="collapsible">
  <h3>Collapsible content header</h3>
  <div data-role="collapsible">
    <h4>Nested Collapsible</h4>
    <p>Nested Collapsible content</p>
  </div>
</div>
```



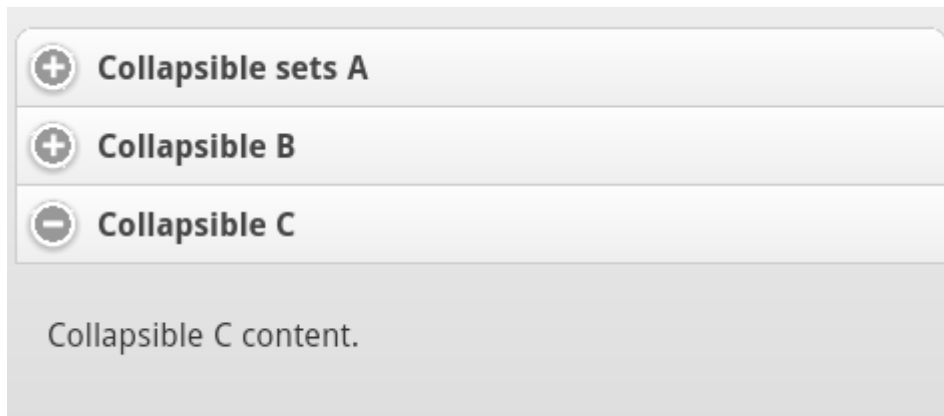
내용을 접을 수 있는 블록 세트(아코디언)(Collapsible sets(accordions))

내용을 접을 수 있는 블록 세트(아코디언)

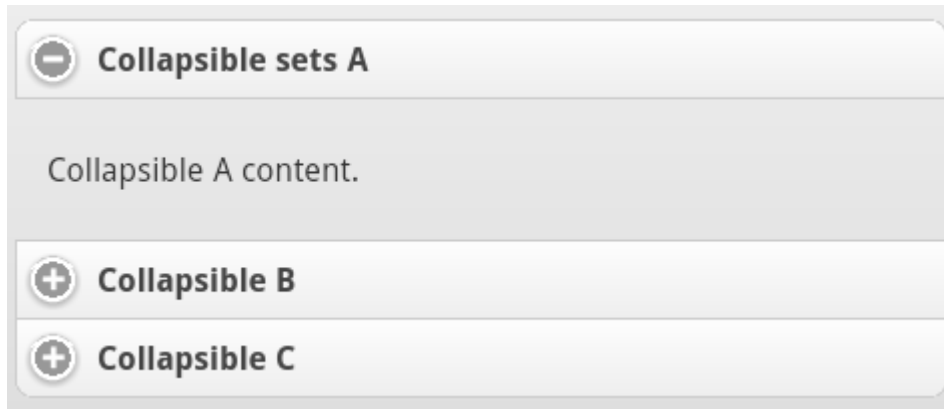
내용을 접을 수 있는 블록을 세트화하여 아코디언 형식으로 만들어줍니다. (내용을 펼치는 경우 펼쳐져 있던 다른 내용을 접어줍니다.) 내용을 접을 수 있는 블록의 컨테이너를 만들어 `data-role="collapsible-set"` 속성을 설정해주어야 합니다.

```
<div data-role="collapsible-set">
  <div data-role="collapsible">
    <h3>Collapsible content header</h3>
    <p>Collapsible content.</p>
  </div>
  <div data-role="collapsible">
    <h3>Collapsible content header</h3>
    <p>Collapsible content.</p>
  </div>
</div>
```

초기 화면



첫 번째 블록을 눌렀을 때 화면



첫 번째 블록을 다시 눌렀을 때 화면

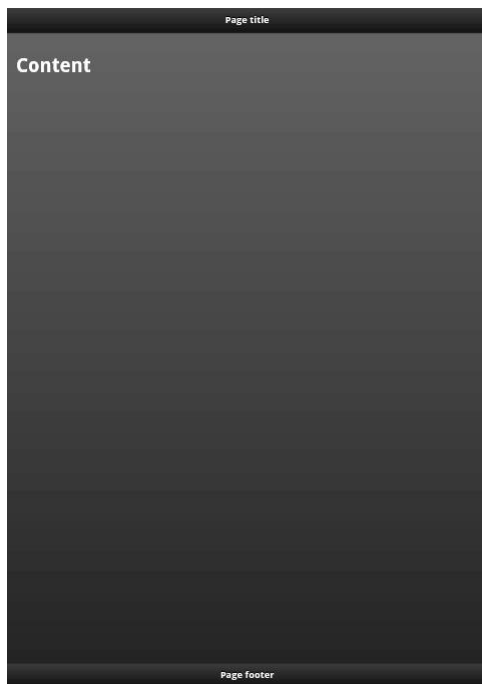


내용 테마(Theming contents)

컨텐츠 영역 테마

컨텐츠 영역에 테마를 적용하기 위해서는 page 의 data-theme 속성을 설정해주면 됩니다.

```
<div data-role="page" data-theme="a">  
    ...  
</div>
```



내용을 접을 수 있는 블록의 테마

data-theme 속성을 지정하여 설정할 수 있습니다.

```
<div data-role="collapsible" data-theme="a">  
    ...  
</div>
```



폼 요소(Form elements)

기본 폼(Form basics)

폼 구조

폼 데이터를 서버에서 처리하기 위해서는 아래와 같은 양식에 쌓여있어야 합니다.

```
<form action="form.php" method="post">
    ...
</form>
```

마크업 규칙

폼 요소는 고유한 아이디를 가지고 있어야 합니다. 여러 페이지에 사용될 수 있기 때문에 주의해야 합니다.

기존 스타일 사용(jQuery Mobile 스타일 사용하지 않기)

특정 양식이 jQuery Mobile 스타일을 사용하지 않기를 원하는 경우 data-role="none" 속성을 설정하면 기존 스타일을 사용할 수 있습니다.

개별적 적용

```
<label for="s">text input</label>  
<input type="text" id="s" data-role="none" />
```

전체적 적용

```
$.mobile.page.prototype.options.keepNavie = "select, input, textarea";
```



폼 요소 스타일 개선

라벨과 넓은 화면의 폼 요소 스타일 개선을 위해 data-role="fieldcontain" 속성을 가진 div 태그나 fieldset 태그를 지정합니다.

```
<div data-role="fieldcontain">  
    ...  
</div>
```

폼 요소 새로 고침

폼 요소를 새로 고침하는 몇 가지 예제입니다. 아래 방법들은 모든 요소를 새로고치기 때문에 새로 고침 방법을 추가로 생각해야 합니다.

체크박스

```
$('#input[type="checkbox"]').attr("checked", true).checkboxradio("refresh");
```

라디오 버튼

```
$('#input[type="radio"]').attr("checked", true).checkboxradio("refresh");
```

셀렉터

```
var selector = $("select");  
selector[0].selectedIndex = 1;  
selector.selectmenu("refresh");
```

슬라이더

```
$('#input[type="range"]').val(50).slider("refresh");
```

토글 스위치

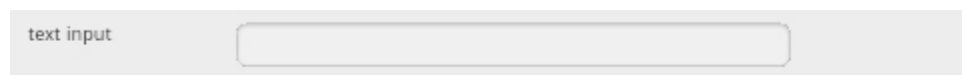
```
var switch = $("select");  
switch[0].selectedIndex = 1;  
switch.slider("refresh");
```

텍스트 입력(Text Inputs)

텍스트 입력

텍스트 입력 요소를 사용하기 위해서는 input 태그와 type="text" 속성을 이용해야 합니다. label 태그의 for 요소를 input 태그의 아이디와 일치하도록 하는것도 중요합니다. 이 요소들은 div 나 fieldset 태그로 둘러싸고 둘러싼 태그에는 data-role="fieldcontainer" 속성을 지정해주어야 합니다.

```
<div data-role="fieldcontain">
  <label for="t">text input</label>
  <input type="text" id="t" />
</div>
```



비밀번호 입력

텍스트 입력을 비밀번호처럼 화면에 표시하지 않기 위해서는 input 요소의 type 을 password 로 변경하면 됩니다.

```
<div data-role="fieldcontain">
  <label for="t">Text Input</label>
  <input type="password" id="t" />
</div>
```



HTML5 표준 입력 유형

HTML5 에서는 숫자, 이메일, URL, 전화번호와 같은 타입을 제공합니다.

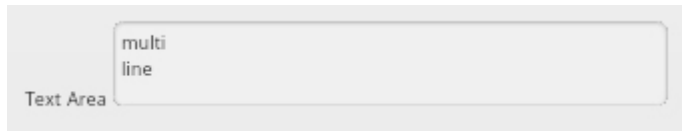
```
<div data-role="fieldcontain">
    <!-- 숫자 -->
    <label for="number">Number</label>
    <input type="number" id="number" />
    <!-- 이메일 -->
    <label for="email">E-mail</label>
    <input type="email" id="email" />
    <!-- URL -->
    <label for="url">URL</label>
    <input type="url" id="url" />
    <!-- 전화번호 -->
    <label for="tel">Tel</label>
    <input type="tel" id="tel" />
</div>
```



텍스트 영역

여러 줄의 텍스트를 입력하기 위해서는 textarea 태그를 사용해야 합니다. textarea 의 높이는 스크롤 바의 필요성을 피하기 위해서 자동으로 조절됩니다.

```
<div data-role="fieldcontain">
    <label for="ta">Text Area</label>
    <textarea id="textarea" id="ta"></textarea>
</div>
```



테마 설정

텍스트 입력 요소의 테마는 data-theme 설정으로 가능합니다.

data-theme 설정

```
<div data-role="fieldcontain">
  <label for="t">text input</label>
  <input type="text" id="t" data-theme="a" />
</div>
```

스크립트를 이용한 설정

```
$("input").textinput({theme: "a"});
```

활성화 및 비활성화

입력을 활성화 및 비활성화 시킬 수 있습니다.

텍스트 입력

```
// 활성화
$("input").textinput("enable");
// 비활성화
$("input").textinput("disable");
```

텍스트 영역

```
// 활성화  
$("#textarea").textinput("enable");  
  
// 비활성화  
$("#textarea").textinput("disable");
```

이벤트

텍스트 입력 관련 이벤트 입니다.

텍스트 변경 이벤트(요소의 입력이 마쳤을 때 텍스트가 변경되었을 경우 발생합니다.)

```
$("#input").bind("change", function (e)  
{  
}
```

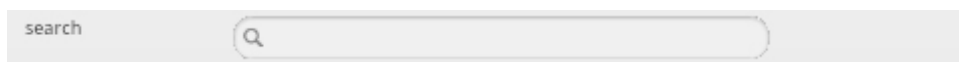
검색 입력(Search inputs)

검색 입력

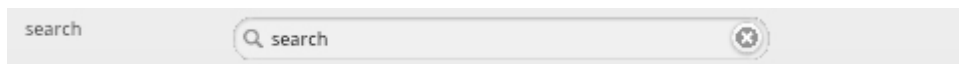
검색 입력 부분은 왼쪽에 search 아이콘이 추가되고 입력시 x 아이콘을 추가합니다. type 을 search 로 설정하면 됩니다.

```
<div data-role="fieldcontain">
  <label for="s">search</label>
  <input type="search" id="s"> </input>
</div>
```

최초 화면



검색어 입력시 화면



input 요소에 data-theme 를 설정하면 테마가 변경됩니다.

```
<div data-role="fieldcontain">
  <label for="s">Text Area</label>
  <input type="search" id="s" data-theme="a"> </input>
</div>
```

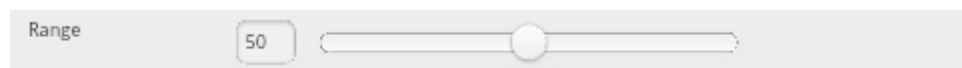


슬라이더(Slider)

슬라이더

슬라이더를 사용하려면 HTML5 의 새로운 유형인 `type="range"` 속성을 지정해야 합니다. 최대값과 최소값을 지정할 수 있으며, 임의 값을 지정해둘 수도 있습니다.

```
<div data-role="fieldcontain">
  <label for="r">Range</label>
  <input type="range" id="r" min="0" max="100" value="50"> </input>
</div>
```



`input` 요소에 `data-theme` 속성을 설정하면 테마가 변경됩니다.

```
<div data-role="fieldcontain">
  <label for="r">Range</label>
  <input type="range" id="r" min="0" max="100" value="50" data-theme="a"> </input>
</div>
```



슬라이더 새로 고침

슬라이더의 값을 설정하고 업데이트 하는 방법입니다.

```
$('#input[type="range"]').val(50).slider("refresh");
```


토글 스위치(Flip toggle switch)

토글 스위치

토글 스위치를 사용하려면 select 요소와 두 가지 옵션을 지정하고 select 에 data-role 속성을 slider 로 설정합니다. 테마 설정은 select 요소에 data-theme 속성을 설정하면 됩니다.

```
<div data-role="fieldcontain">
  <label for="ts">Flip toggle switch</label>
  <select data-role="slider" id="ts">
    <option value="off">off</option>
    <option value="on">on</option>
  </select>
</div>
```

최초 화면



off 를 눌렀을 때 화면



테마 적용 화면

```
<div data-role="fieldcontain">
  <label for="ts">Flip toggle switch</label>
  <select data-role="slider" id="ts" data-theme="a">
    <option value="off">off</option>
    <option value="on">on</option>
  </select>
</div>
```



토글 스위치 새로 고침

토글 스위치를 업데이트하고 새로 고침 하는 방법입니다.

```
var switch = $("select");  
switch[0].selectedIndex = 1;  
switch[0].slider("refresh");
```

라디오 버튼(Radio buttons)

라디오 버튼

라디오 버튼은 하나의 항목을 선택할 수 있습니다.

세로로 그룹 된 라디오 버튼

라디오 버튼을 만들려면 input 요소의 type 속성을 radio 로 설정합니다. 버튼이 같은 그룹이라는 것을 설정하기 위해 fieldset 태그로 요소를 감싸고 data-role="controlgroup" 속성을 설정합니다. 그리고 같은 그룹 요소는 input 태그의 name 을 같게 설정해야 합니다.

```
<div data-role="fieldcontain">
  <fieldset data-role="controlgroup">
    <label for="r0">Radio 00</label>
    <input type="radio" name="rg0" id="r0" value="0" />
    <label for="r1">Radio 01</label>
    <input type="radio" name="rg0" id="r1" value="1" />
    <label for="r2">Radio 02</label>
    <input type="radio" name="rg0" id="r2" value="2" />
  </fieldset>
</div>
```

최초 화면



Radio 00 선택 화면



A vertical group of three radio buttons. The first button, labeled "Radio 00", is selected and has a blue dot. The second button, labeled "Radio 01", and the third button, labeled "Radio 02", are unselected and have grey dots. The buttons are contained within a light grey rounded rectangle.

Radio 01 선택 화면



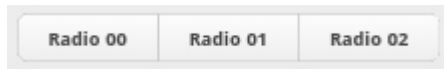
A vertical group of three radio buttons. The second button, labeled "Radio 01", is selected and has a blue dot. The first button, labeled "Radio 00", and the third button, labeled "Radio 02", are unselected and have grey dots. The buttons are contained within a light grey rounded rectangle.

가로로 그룹 된 라디오 버튼

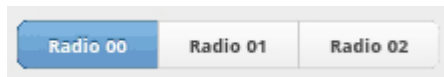
fieldset 요소의 data-type="horizontal" 로 설정하면 라디오 버튼이 가로로 나열됩니다.

```
<div data-role="fieldcontain">
  <fieldset data-role="controlgroup" data-type="horizontal">
    <label for="r0">Radio 00</label>
    <input type="radio" name="rg0" id="r0" value="0" />
    <label for="r1">Radio 01</label>
    <input type="radio" name="rg0" id="r1" value="1" />
    <label for="r2">Radio 02</label>
    <input type="radio" name="rg0" id="r2" value="2" />
  </fieldset>
</div>
```

최초 화면



Radio 00 선택 화면



Radio 01 선택 화면



테마 설정

텍스트 입력 요소의 테마는 data-theme 설정으로 가능합니다.

data-theme 설정

```
<div data-role="fieldcontain">
  <fieldset data-role="controlgroup">
    <label for="r0">Radio 00</label>
    <input type="radio" name="rg0" id="r0" value="0" data-theme="a" />
    <label for="r1">Radio 01</label>
    <input type="radio" name="rg0" id="r1" value="1" data-theme="b" />
    <label for="r2">Radio 02</label>
    <input type="radio" name="rg0" id="r2" value="2" data-theme="c" />
  </fieldset>
</div>
```

스크립트를 이용한 설정

```
$("input[type='radio']").checkboxradio({theme: "a"});
```

활성화 및 비활성화

라디오 버튼 선택을 활성화 및 비활성화할 수 있습니다.

```
// 활성화
$("input[type='radio']").checkboxradio("enable");
// 비활성화
$("input[type='radio']").checkboxradio("disable");
```

라디오 버튼 새로 고침

라디오 버튼을 아무것도 선택하지 않은 상태로 새로 고침 하는 방법입니다.

```
$("input[type='radio']").attr("checked", false).checkboxradio("refresh");
```

이벤트

라디오 버튼 관련 이벤트 입니다.

라디오 버튼의 선택이 변경되는 경우 발생합니다.

```
$("input[type='radio']").bind("change", function (e, ui)
{
}
```

체크박스(Checkboxes)

체크박스

체크박스는 하나 이상의 요소를 선택할 수 있습니다. input 요소의 type 속성을 checkbox 로 설정하고 요소들을 fieldset 태그로 감싸고 fieldset 에 data-role="controlgroup" 속성을 설정합니다. input 요소의 name 속성은 다르게 설정해야 합니다. 같게 설정하면 라디오버튼과 같게 동작합니다.

```
<div data-role="fieldcontain">
  <fieldset data-role="controlgroup">
    <label for="cb0">Checkbox 00</label>
    <input type="checkbox" name="cbg0" id="cb0" value="0" />
    <label for="cb1">Checkbox 01</label>
    <input type="checkbox" name="cbg1" id="cb1" value="1" />
    <label for="cb2">Checkbox 02</label>
    <input type="checkbox" name="cbg2" id="cb2" value="2" />
  </fieldset>
</div>
```

최초 화면



A screenshot of a mobile application interface showing a list of three checkboxes. Each checkbox is represented by a small square icon followed by the text label. All three checkboxes are currently unchecked.

Checkbox 00
<input type="checkbox"/>
Checkbox 01
<input type="checkbox"/>
Checkbox 02
<input type="checkbox"/>

Checkbox 00 선택 화면



A screenshot of the same mobile application interface, but now the first checkbox, 'Checkbox 00', is checked. The checkbox icon is now a blue square with a white checkmark. The other two checkboxes remain unchecked.

Checkbox 00
<input checked="" type="checkbox"/>
Checkbox 01
<input type="checkbox"/>
Checkbox 02
<input type="checkbox"/>

Checkbox 00 과 Checkbox 01 선택 화면

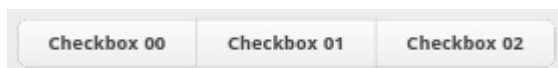


가로로 그룹된 체크박스

체크박스를 가로로 나열하기 위해서는 fieldset 에 data-type="horizontal" 을 설정합니다.

```
<div data-role="fieldcontain">
  <fieldset data-role="controlgroup" data-type="horizontal">
    <label for="cb0">Checkbox 00</label>
    <input type="checkbox" name="cbg0" id="cb0" value="0" />
    <label for="cb1">Checkbox 01</label>
    <input type="checkbox" name="cbg1" id="cb1" value="1" />
    <label for="cb2">Checkbox 02</label>
    <input type="checkbox" name="cbg2" id="cb2" value="2" />
  </fieldset>
</div>
```

최초 화면



Checkbox 00 선택 화면



Checkbox 00 과 Checkbox 01 선택 화면



테마 설정

텍스트 입력 요소의 테마는 data-theme 설정으로 가능합니다.

data-theme 설정

```
<div data-role="fieldcontain">
  <fieldset data-role="controlgroup">
    <label for="cb0">Checkbox 00</label>
    <input type="checkbox" name="cbg0" id="cb0" value="0" data-theme="a" />
    <label for="cb1">Checkbox 01</label>
    <input type="checkbox" name="cbg1" id="cb1" value="1" data-theme="b" />
    <label for="cb2">Checkbox 02</label>
    <input type="checkbox" name="cbg2" id="cb2" value="2" data-theme="c" />
  </fieldset>
</div>
```

스크립트를 이용한 설정

```
$("input[type='checkbox']").checkboxradio({theme: "a"});
```

활성화 및 비활성화

체크박스 선택을 활성화 및 비활성화할 수 있습니다.

```
// 활성화
```

```
$("#input[type='checkbox']").checkboxradio("enable");  
// 비활성화  
$("#input[type='checkbox']").checkboxradio("disable");
```

라디오 버튼 새로 고침

라디오 버튼을 아무것도 선택하지 않은 상태로 새로 고침 하는 방법입니다.

```
$('#input[type="checkbox"]').attr("checked", false).checkboxradio("refresh");
```

이벤트

라디오 버튼 관련 이벤트 입니다.

라디오 버튼의 선택이 변경되는 경우 발생합니다.

```
$("#input[type='radio']").bind("change", function (e, ui)  
{  
}
```

선택 메뉴(Select menus)

선택 메뉴

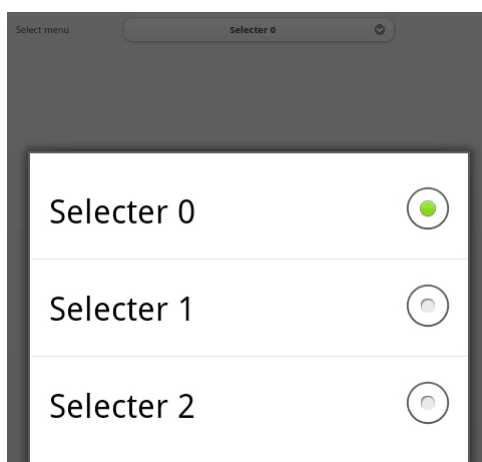
콤보박스나 셀렉터라고도 많이 부르는 선택 메뉴 입니다. data-role="fieldcontain" 으로 설정된 div 안에 select 를 만들어주면 됩니다.

```
<div data-role="fieldcontain">
  <label for="s">Select menu</label>
  <select id="s">
    <option value="0">Selector 0</option>
    <option value="1">Selector 1</option>
    <option value="2">Selector 2</option>
  </select>
</div>
```

최초 화면



선택 메뉴 터치시 화면



Selector 1 선택 화면



커스텀 메뉴

jQuery Mobile 에서 제공되는 메뉴를 사용하면 다양한 플랫폼과 테마에 대한 시각적인 일관성을 제공합니다. 선택 옵션이 많은 경우 대화창으로 보여집니다.

사이트 전체에 공통되게 적용할 때는 mobileinit 이벤트를 활용하여 설정할 수 있습니다.

```
$(document).bind("mobileinit", function ()
{
    $.mobile.selectmenu.prototype.options.nativeMenu = false;
});
```

select 에 data-native-menu="false" 로 설정주어도 사용할 수 있습니다.

data-native-menu 속성

```
<div data-role="fieldcontain">
    <label for="s">Select menu</label>
    <select id="s" data-native-menu="false">
        <option value="0">Selector 0</option>
        <option value="1">Selector 1</option>
        <option value="2">Selector 2</option>
    </select>
</div>
```

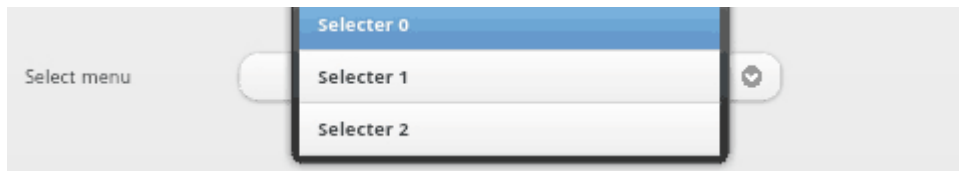
스크립트

```
$("select").selectmenu({nativeMenu:false});
```

최초 화면



선택 메뉴 터치시 화면



Selector 1 선택 화면



자리 표시 옵션

선택 메뉴 안에 메뉴에 대한 텍스트를 표시해줍니다.

selectmenu 플러그인을 통해 기능을 비활성화 할 수 있습니다.

```
$(document).bind("mobileinit", function ()  
{  
    $.mobile.selectmenu.prototype.options.hidePlaceholderMenuItems = false;  
});
```

자리 표시자를 만드는 방법은 아래와 같습니다.

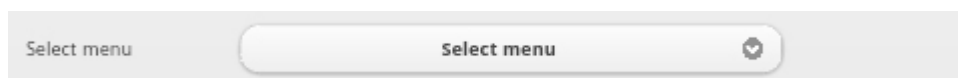
```
<div data-role="fieldcontain">  
    <label for="s">Select menu</label>
```

```

<select id="s" data-native-menu="false">
  <option value="title" data-placeholder="true">Select menu</option>
  <option value="0">Selector 0</option>
  <option value="1">Selector 1</option>
  <option value="2">Selector 2</option>
</select>
</div>

```

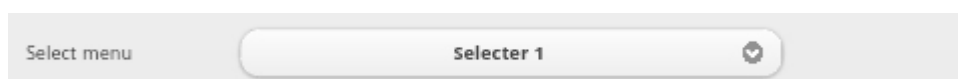
최초 화면



선택 메뉴 터치시 화면



Selector 1 선택 화면



요소 비활성화

선택 메뉴의 요소에 `disabled="true"` 속성을 주면 요소를 선택할 수 없게 비활성화 합니다.

```

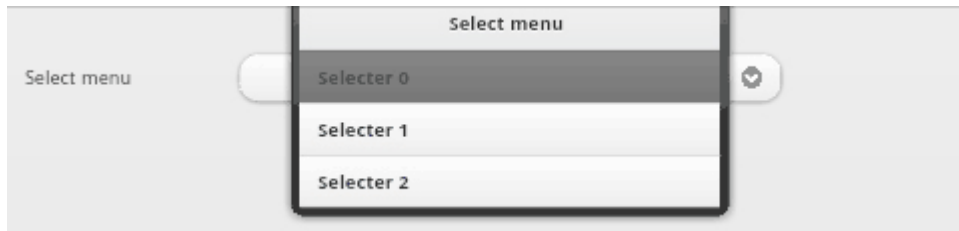
<div data-role="fieldcontain">
  <label for="s">Select menu</label>
  <select id="s" data-native-menu="false">
    <option value="title" data-placeholder="true">Select menu</option>

```

```

        <option value="0" disabled="true">Selector 0</option>
        <option value="1">Selector 1</option>
        <option value="2">Selector 2</option>
    </select>
</div>

```



그룹 지원

선택 메뉴 요소에 그룹을 지어줄 수 있습니다.

optgroup 를 이용해 요소를 그룹지어주면 됩니다. optgroup 의 label 속성을 설정해주면 그룹명으로 표시됩니다.

```

<div data-role="fieldcontain">
    <label for="s">Select menu</label>
    <select id="s" data-native-menu="false">
        <option value="title" data-placeholder="true">Select menu</option>
        <optgroup label="First group">
            <option value="0">Selector 00</option>
            <option value="1">Selector 01</option>
            <option value="2">Selector 02</option>
        </optgroup>
        <optgroup label="Second group">
            <option value="3">Selector 10</option>
            <option value="4">Selector 11</option>
            <option value="5">Selector 12</option>
        </optgroup>
    </select>
</div>

```

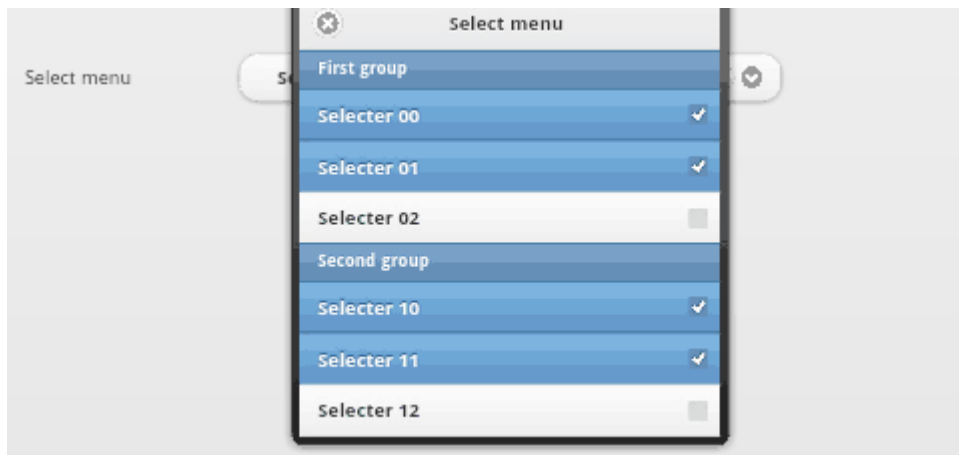


복수 선택

선택 메뉴의 요소를 복수로 선택하도록 할 수 있습니다. select 요소에 `multiple="true"` 속성을 지정해주면 선택메뉴 터치시 닫기 버튼이 나오고 요소들 오른쪽에 체크 표시가 되는 영역이 나타납니다. 선택을 하고 선택 메뉴를 닫으면 선택한 메뉴가 콤마(,)로 분류되 표시되고 오른쪽에 선택한 갯수가 보여집니다.

```
<div data-role="fieldcontain">
  <label for="s">Select menu</label>
  <select id="s" data-native-menu="false" multiple="true">
    <option value="title" data-placeholder="true">Select menu</option>
    <optgroup label="First group">
      <option value="0">Selector 00</option>
      <option value="1">Selector 01</option>
      <option value="2">Selector 02</option>
    </optgroup>
    <optgroup label="Second group">
      <option value="3">Selector 10</option>
      <option value="4">Selector 11</option>
      <option value="5">Selector 12</option>
    </optgroup>
  </select>
</div>
```


Selector 00, Select 01, Selector 10, Selector 11 선택 화면



데이터 속성 지원

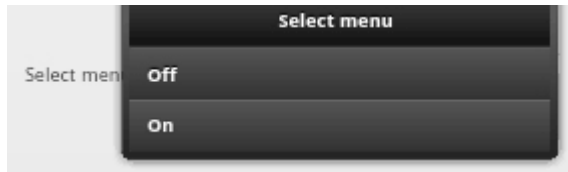
선택 메뉴는 데이터 속성을 지원하기 합니다. 아래는 버튼처럼 보이도록 간단하게 적용해본 내용입니다.

```
<div data-role="fieldcontain">
  <label for="s">Select menu</label>
  <select id="s" data-native-menu="false" data-inline="true" data-icon="gear" data-theme="a">
    <option value="off">Off</option>
    <option value="on">On</option>
  </select>
</div>
```

최초 화면



터치시 화면



둥근 모서리 변경

data-corners 속성을 false 로 설정하면 둥근 모서리가 각진 모서리로 변경됩니다.

data-corners 속성 설정

```
<div data-role="fieldcontain">
  <label for="s">Select menu</label>
  <select id="s" data-native-menu="false" data-corners="false">
    <option value="0">Selector 0</option>
    <option value="1">Selector 1</option>
    <option value="2">Selector 2</option>
  </select>
</div>
```

스크립트

```
$("select").selectmenu({corners:false});
```



아이콘 변경

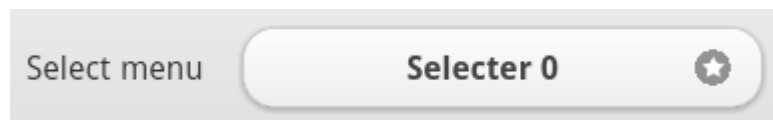
선택 메뉴의 아이콘은 data-icon 속성을 통해 변경 가능합니다. 기본 아이콘은 arrow-d 로 설정되어 있습니다.

data-icon 속성 설정

```
<div data-role="fieldcontain">
  <label for="s">Select menu</label>
  <select id="s" data-native-menu="false" data-icon="star">
    <option value="0">Selector 0</option>
    <option value="1">Selector 1</option>
    <option value="2">Selector 2</option>
  </select>
</div>
```

스크립트

```
$("select").selectmenu({icon:"star"});
```



아이콘 위치 설정

선택 메뉴의 아이콘 위치를 data-iconpos 속성을 이용해 변경 가능합니다. 기본은 right 입니다.

data-iconpos 속성 설정

```
<div data-role="fieldcontain">
  <label for="s">Select menu</label>
  <select id="s" data-native-menu="false" data-iconpos="left">
    <option value="0">Selector 0</option>
    <option value="1">Selector 1</option>
  </select>
</div>
```

```

        <option value="2">Selector 2</option>
    </select>
</div>

```

스크립트

```

$("select").selectmenu({iconpos:"left"});

```



아이콘 그림자 설정

선택 메뉴의 아이콘 그림자를 data-iconshadow 속성을 이용해 변경 가능합니다. 기본은 true 입니다.

data-iconshadow 속성 설정

```

<div data-role="fieldcontain">
    <label for="s">Select menu</label>
    <select id="s" data-native-menu="false" data-iconshadow="false">
        <option value="0">Selector 0</option>
        <option value="1">Selector 1</option>
        <option value="2">Selector 2</option>
    </select>
</div>

```

스크립트

```

$("select").selectmenu({iconshadow:"false"});

```

inline 스타일 설정

선택 메뉴의 아이콘을 inline 스타일로 변경하기 위해서는 data-inline 속성을 설정해야 합니다. 기본값은 false 입니다.

data-inline 속성 설정

```
<div data-role="fieldcontain">
  <label for="s">Select menu</label>
  <select id="s" data-native-menu="false" data-inline="true">
    <option value="0">Selector 0</option>
    <option value="1">Selector 1</option>
    <option value="2">Selector 2</option>
  </select>
</div>
```

스크립트

```
$("select").selectmenu({inline:true});
```



그림자 설정

선택 메뉴의 그림자에 대한 설정을 변경할 수 있습니다. 기본으로는 그림자가 있습니다.

data-shadow 속성 설정

```
<div data-role="fieldcontain">
  <label for="s">Select menu</label>
  <select id="s" data-native-menu="false" data-shadow="false">
    <option value="0">Selector 0</option>
    <option value="1">Selector 1</option>
  </select>
</div>
```

```

        <option value="2">Selector 2</option>
    </select>
</div>

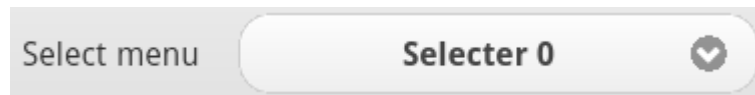
```

스크립트

```

$("select").selectmenu({shadow:false});

```



테마 설정

선택 메뉴의 테마를 변경할 수 있습니다.

data-theme 속성 설정

```

<div data-role="fieldcontain">
    <label for="s">Select menu</label>
    <select id="s" data-native-menu="false" data-theme="a">
        <option value="0">Selector 0</option>
        <option value="1">Selector 1</option>
        <option value="2">Selector 2</option>
    </select>
</div>

```

스크립트

```

$("select").selectmenu({theme:"a"});

```

선택 메뉴 열기 및 닫기

스크립트를 이용해 선택 메뉴를 열고 닫는 방법입니다.

```
// 열기
$("select").selectmenu("open");
// 닫기
$("select").selectmenu("close");
```

활성화 및 비활성화

선택 메뉴의 사용을 활성화 및 비활성화로 설정할 수 있습니다.

```
// 활성화
$("select").selectmenu("enable");
// 비활성화
$("select").selectmenu("disable");
```

새로 고침

선택 메뉴를 업데이트하고 새로 고침하는 방법입니다.

```
var select = $("select");
select[0].selectedIndex = 0;
select[0].selectmenu("refresh");
```

이벤트

선택 메뉴 관련 이벤트 입니다.

선택 메뉴의 선택이 변경되는 경우 발생합니다.

```
$("#select").bind("change", function (e, ui)
{
}
```


폼 테마(Theming forms)

폼 테마

폼 요소의 테마는 요소에 직접 data-theme 를 설정하거나 컨테이너에 data-theme 를 설정해주면 컨테이너 내의 요소들의 테마가 공통적으로 적용됩니다. (1.0 beta 2 로 버전이 업데이트 되면서 일일이 적용해주어야 하는 요소들이 있습니다.)

관련 메서드(Plugin methods)

메서드 전체적으로 활성화/비활성화 관련 내용들은 제대로 동작하지 않는 것으로 보입니다.

선택 메뉴

선택 메뉴 열기

```
$("#select").selectmenu("open");
```

선택 메뉴 닫기

```
$("#select").selectmenu("close");
```

선택 메뉴 새로 고침

```
$("#select").selectmenu("refresh");
```

선택 메뉴 다시 만들기

```
$("#select").selectmenu("refresh", true);
```

선택 메뉴 활성화/비활성화

```
// 활성화
$("#select").selectmenu("enable");
// 비활성화
$("#select").selectmenu("disable");
```

텍스트 입력

```
// 활성화
$("input").textinput("enable");
// 비활성화
$("input").textinput("disable");
```

텍스트 영역

```
// 활성화
$("textarea").textinput("enable");
// 비활성화
$("textarea").textinput("disable");
```

체크박스/라디오버튼

```
// 활성화
$("input").checkboxradio("enable");
// 비활성화
$("input").checkboxradio("disable");
```

체크박스/라디오버튼 새로 고침

```
$("input").checkboxradio("refresh");
```

슬라이더

슬라이더 활성화/비활성화 (버그인지 적용이 되지 않습니다.)

```
// 활성화  
$("input").slider("enable");  
// 비활성화  
$("input").slider("disable");
```

슬라이더 새로 고침

```
$("input").slider("refresh");
```

폼 버튼

폼 버튼 활성화/비활성화 (버그인지 적용이 되지 않습니다.)

```
// 활성화  
$("input").button("enable");  
// 비활성화  
$("input").button("disable");
```

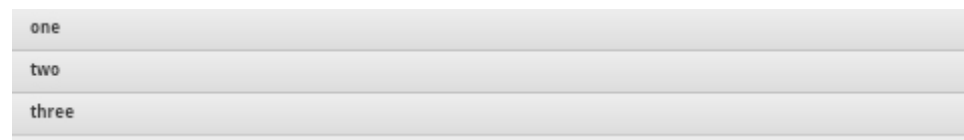
리스트(List views)

읽기 전용 리스트(Read-only list)

읽기 전용 리스트

읽기 전용 리스트는 정렬되지 않은 리스트로 작성되고 data-role="listview" 속성을 지정해야 합니다.

```
<ul data-role="listview">  
  <li>one</li>  
  <li>two</li>  
  <li>three</li>  
</ul>
```



기본 링크 리스트(Basic linked list)

기본 링크 리스트

기본적인 리스트는 리스트 아이템을 a 태그로 감싸주면 생성됩니다. 브라우저의 전체 너비로 채워지며 오른쪽에 화살표 표시가 나타납니다.

```
<ul data-role="listview">
  <li>
    <a href="one.html">one</a>
  </li>
  <li>
    <a href="two.html">two</a>
  </li>
  <li>
    <a href="three.html">three</a>
  </li>
</ul>
```



중첩 목록(Nested list)

중첩 목록

li 요소 안에 새로운 리스트를 만들어주면 중첩된 리스트를 만들수 있습니다.

```
<ul data-role="listview">
  <li>
    <a href="one.html">one</a>
    <ul>
      <li><a href="onenone.html">One-One</a></li>
      <li><a href="onentwo.html">One-Two</a></li>
    </ul>
  </li>
  <li>
    <a href="two.html">two</a>
    <ul>
      <li><a href="twonone.html">Two-One</a></li>
      <li><a href="twontwo.html">Two-Two</a></li>
    </ul>
  </li>
</ul>
```

최초 화면

one	▶
two	▶

리스트 항목 중 one 터치 후 화면

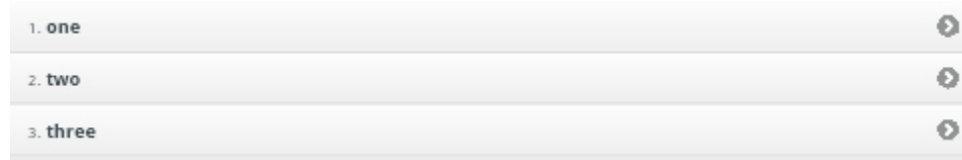
one	
One-One	▶
One-Two	▶

번호가 붙은 리스트(Numbered list)

번호가 붙은 리스트

ol 태그로 리스트를 만들어주면 리스트 목록에 번호가 붙게 됩니다.

```
<ol data-role="listview">
  <li>
    <a href="one.html">one</a>
  </li>
  <li>
    <a href="two.html">two</a>
  </li>
  <li>
    <a href="three.html">three</a>
  </li>
</ol>
```



버튼이 나뉜 리스트(Split button list)

버튼이 나뉜 리스트

리스트 항목이 하나 이상의 행동이 가능하게 해야 하는 경우 버튼이 나뉜 리스트를 이용하면 터치할 수 있는 두 가지 항목이 제공됩니다. a 태그를 두 가지 지정해주면 처음의 a 태그는 리스트 항목 내용으로 두 번째 태그는 아이콘 부분을 담당합니다. 나뉜 항목의 아이콘을 설정하기 위해서는 리스트에 data-split-icon 속성을 설정해야 합니다.

```
<ul data-role="listview" data-split-icon="gear">
  <li>
    <a href="one.html">one</a>
    <a href="aone.html">one</a>
  </li>
  <li>
    <a href="two.html">two</a>
    <a href="atwo.html">two</a>
  </li>
  <li>
    <a href="three.html">three</a>
    <a href="athree.html">three</a>
  </li>
</ul>
```

one	
two	
three	

리스트 구분자(List dividers)

리스트 구분자

리스트 항목에 구분자를 만들기 위해서는 리스트 항목에 `data-role="list-divider"` 속성을 추가해주어야 합니다.

```
<ul data-role="listview">
  <li data-role="list-divider">First divider</li>
  <li>
    <a href="one.html">one</a>
  </li>
  <li data-role="list-divider">Second divider</li>
  <li>
    <a href="two.html">two</a>
  </li>
  <li data-role="list-divider">Third divider</li>
  <li>
    <a href="three.html">three</a>
  </li>
</ul>
```



리스트 구분자의 테마 설정은 리스트 요소에 `data-dividertheme` 속성을 설정해야 합니다.

```
<ul data-role="listview" data-dividertheme="a">
  <li data-role="list-divider">First divider</li>
  <li>
    <a href="one.html">one</a>
  </li>
```

```
<li data-role="list-divider">Second divider</li>
<li>
    <a href="two.html">two</a>
</li>
<li data-role="list-divider">Third divider</li>
<li>
    <a href="three.html">three</a>
</li>
</ul>
```



카운터 풍선(Count bubble)

카운터 풍선

리스트의 오른쪽에 카운터를 보여주는 역할을 합니다. 카운터 풍선을 만들기 위해서는 ui-li-count 클래스를 적용해 주어야 합니다.

```
<ul data-role="listview">
  <li>
    <a href="one.html">
      <h3>one</h3>
      <span class="ui-li-count">1</span>
    </a>
  </li>
  <li>
    <a href="two.html">
      <h3>two</h3>
      <span class="ui-li-count">2</span>
    </a>
  </li>
  <li>
    <a href="three.html">
      <h3>three</h3>
      <span class="ui-li-count">3</span>
    </a>
  </li>
</ul>
```



썸네일 리스트(Thumbnails)

썸네일 리스트

썸네일 리스트를 사용하면 리스트 항목의 왼쪽에 이미지를 보여줄 수 있습니다. 썸네일 이미지의 권장 크기는 가로세로 80px 입니다. 리스트 항목에 `img` 태그를 이용해 이미지를 추가해주면 됩니다.

```
<ul data-role="listview">
  <li>
    <a href="one.html">
      
      <h3>one</h3>
    </a>
  </li>
  <li>
    <a href="two.html">
      
      <h3>two</h3>
    </a>
  </li>
  <li>
    <a href="three.html">
      
      <h3>three</h3>
    </a>
  </li>
</ul>
```



아이콘 리스트(Icons)

아이콘 리스트

썸네일 리스트와 같은 방법으로 만들어 `img` 태그에 `ui-li-icon` 클래스를 설정해주면 됩니다. 권장 이미지 크기는 가로세로 16px 입니다.

```
<ul data-role="listview">
  <li>
    <a href="one.html">
      
      <span>one</span>
    </a>
  </li>
  <li>
    <a href="two.html">
      
      <span>two</span>
    </a>
  </li>
  <li>
    <a href="three.html">
      
      <span>three</span>
    </a>
  </li>
</ul>
```



내용 설정(Content formatting)

부 제목

리스트 항목에 부 제목을 넣어주기 위해서는 제목 이외에 p 태그를 이용해 내용을 작성해주면 됩니다.

```
<ul data-role="listview">
  <li>
    <a href="one.html">
      <h3>one</h3>
      <p>One detail</p>
    </a>
  </li>
  <li>
    <a href="two.html">
      <h3>two</h3>
      <p>Two detail</p>
    </a>
  </li>
  <li>
    <a href="three.html">
      <h3>three</h3>
      <p>Three detail</p>
    </a>
  </li>
</ul>
```



부 제목 추가

리스트 항목에 표현할 내용이 하나로 부족한 경우 내용 한 부분을 진하게 설정해 표시할 수 있습니다.

```
<ul data-role="listview">
  <li>
    <a href="one.html">
      <h3>one</h3>
      <p><strong>One detail</strong></p>
      <p>One description</p>
    </a>
  </li>
  <li>
    <a href="two.html">
      <h3>two</h3>
      <p><strong>Two detail</strong></p>
      <p>Two description</p>
    </a>
  </li>
  <li>
    <a href="three.html">
      <h3>three</h3>
      <p><strong>Three detail</strong></p>
      <p>Three description</p>
    </a>
  </li>
</ul>
```



리스트 항목 오른쪽 내용

리스트 항목의 오른쪽 부분에 글을 표시하기 위해서는 p 태그에 ui-li-aside 클래스를 추가해주면 됩니다. 작성한 시간 같은 내용을 표시할 때 유용합니다.

```
<ul data-role="listview">
  <li>
    <a href="one.html">
      <h3>one</h3>
      <p><strong>One detail</strong></p>
      <p>One description</p>
      <p class="ui-li-aside">One write time</p>
    </a>
  </li>
  <li>
    <a href="two.html">
      <h3>two</h3>
      <p><strong>Two detail</strong></p>
      <p>Two description</p>
      <p class="ui-li-aside">Two write time</p>
    </a>
  </li>
  <li>
    <a href="three.html">
      <h3>three</h3>
      <p><strong>Three detail</strong></p>
      <p>Three description</p>
      <p class="ui-li-aside">Three write time</p>
    </a>
  </li>
</ul>
```

one One detail One description	One write time	▶
two Two detail Two description	Two write time	▶
three Three detail Three description	Three write time	▶

리스트 검색 바(Search filter bar)

리스트 검색 바

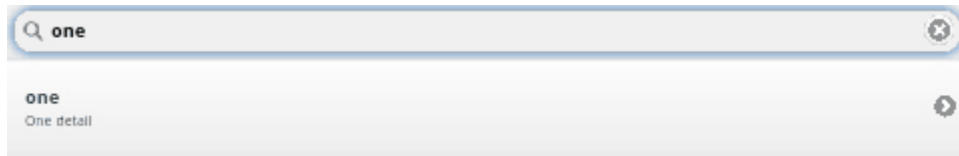
jQuery Mobile 은 클라이언트에서 리스트를 검색하는 아주 쉬운 방법을 제공합니다. 리스트에 검색 바를 추가하려면 data-filter="true" 속성을 리스트에 추가해주면 됩니다.

```
<ul data-role="listview" data-filter="true">
  <li>
    <a href="one.html">
      <h3>one</h3>
      <p>One detail</p>
    </a>
  </li>
  <li>
    <a href="two.html">
      <h3>two</h3>
      <p>Two detail</p>
    </a>
  </li>
  <li>
    <a href="three.html">
      <h3>three</h3>
      <p>Three detail</p>
    </a>
  </li>
</ul>
```

최초 화면



검색어로 one 입력시 화면



검색 바에 표시되는 문구를 변경하기 위해서는 mobile 이벤트를 사용하여 설정을 바꿔주어야 합니다.

```
$mobile.listview.prototype.options.filterPlaceholder = "Search...";
```

테마 변경은 data-filter-theme 속성을 설정하면 변경됩니다.

```
<ul data-role="listview" data-filter="true" data-filter-theme="a">
  <li>
    <a href="one.html">
      <h3>one</h3>
      <p>One detail</p>
    </a>
  </li>
  <li>
    <a href="two.html">
      <h3>two</h3>
      <p>Two detail</p>
    </a>
  </li>
  <li>
    <a href="three.html">
      <h3>three</h3>
      <p>Three detail</p>
    </a>
  </li>
</ul>
```

Filter items...	
one One detail	➤
two Two detail	➤
three Three detail	➤

안으로 들어간 리스트 검색 바(Inset search filter bar)

안으로 들어간 리스트 검색 바

리스트에 `data-inset="true"` 속성을 설정하면 페이지 내부로 리스트가 들어가 보이게 표현됩니다.
검색 바가 없어도 똑같이 표현됩니다.

```
<ul data-role="listview" data-filter="true" data-inset="true">
  <li>
    <a href="one.html">
      <h3>one</h3>
      <p>One detail</p>
    </a>
  </li>
  <li>
    <a href="two.html">
      <h3>two</h3>
      <p>Two detail</p>
    </a>
  </li>
  <li>
    <a href="three.html">
      <h3>three</h3>
      <p>Three detail</p>
    </a>
  </li>
</ul>
```



구분자가 들어있는 리스트 검색 바(Search filter bar with dividers)

구분자가 들어있는 리스트 검색 바

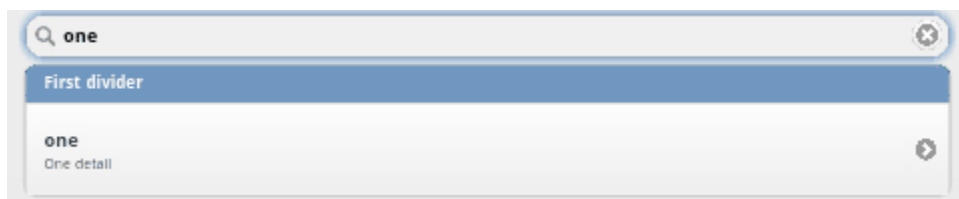
구분자가 있는 리스트에 검색 바가 추가되면 검색했을 때 구분자와 관련된 항목이 남아있는 경우 항목 위에 구분자가 남아서 표시됩니다.

```
<ul data-role="listview" data-filter="true" data-inset="true">
  <li data-role="list-divider">First divider</li>
  <li>
    <a href="one.html">
      <h3>one</h3>
      <p>One detail</p>
    </a>
  </li>
  <li data-role="list-divider">Second divider</li>
  <li>
    <a href="two.html">
      <h3>two</h3>
      <p>Two detail</p>
    </a>
  </li>
  <li data-role="list-divider">Third divider</li>
  <li>
    <a href="three.html">
      <h3>three</h3>
      <p>Three detail</p>
    </a>
  </li>
</ul>
```

초기 화면



검색어로 one 입력시 화면



리스트 테마(Theming lists)

리스트 테마

리스트에 테마를 설정하기 위해서는 리스트 요소에 data-theme 속성을 설정해주어야 합니다.

```
<ul data-role="listview" data-theme="a">
  <li>one</li>
  <li>two</li>
  <li>three</li>
</ul>
```



리스트 구분자 테마

리스트 구분자에 테마를 설정하기 위해서는 리스트 요소에 data-divider-theme 속성을 설정해주어야 합니다. (data-divider-theme, data-dividertHEME 둘 다 가능합니다.)

```
<ul data-role="listview" data-divider-theme="a">
  <li data-role="list-divider">First divider</li>
  <li>one</li>
  <li data-role="list-divider">Second divider</li>
  <li>two</li>
  <li data-role="list-divider">Third divider</li>
  <li>three</li>
</ul>
```


First divider
one
Second divider
two
Third divider
three

카운터 풍선 테마

카운터 풍선의 테마는 리스트 요소에 data-count-theme 속성을 설정하는 것으로 변경 가능합니다.

```
<ul data-role="listview" data-count-theme="a">
  <li>
    <a href="one.html">
      <h3>one</h3>
      <span class="ui-li-count">1</span>
    </a>
  </li>
  <li>
    <a href="two.html">
      <h3>two</h3>
      <span class="ui-li-count">2</span>
    </a>
  </li>
  <li>
    <a href="three.html">
      <h3>three</h3>
      <span class="ui-li-count">3</span>
    </a>
  </li>
</ul>
```



기본적인 링크되는 리스트 아이콘 변경

아이콘을 변경하기 위해선 리스트 항목에 `data-icon` 속성을 설정해주어야 합니다. `false` 값을 설정 해주면 아이콘이 표시되지 않습니다.

```
<ul data-role="listview">
  <li data-icon="arrow-l">
    <a href="one.html">one</a>
  </li>
  <li data-icon="false">
    <a href="two.html">two</a>
  </li>
  <li data-icon="false">
    <a href="three.html">three</a>
  </li>
</ul>
```



버튼이 나뉜 리스트 테마

나누어진 부분의 테마를 변경하기 위해서는 리스트에 `data-split-theme` 속성을 설정해주어야 합니다.

```
<ul data-role="listview" data-split-theme="a">
  <li>
    <a href="one.html">one</a>
    <a href="aone.html">one</a>
  </li>
  <li>
    <a href="two.html">two</a>
    <a href="atwo.html">two</a>
  </li>
  <li>
    <a href="three.html">three</a>
    <a href="athree.html">three</a>
  </li>
</ul>
```



버튼이 나뉜 리스트 아이콘 변경

아이콘 변경을 위해서는 리스트에 `data-split-icon` 을 설정하면 변경됩니다.

```
<ul data-role="listview" data-split-icon="delete">
  <li>
    <a href="one.html">one</a>
    <a href="aone.html">one</a>
  </li>
  <li>
    <a href="two.html">two</a>
```

```
                <a href="atwo.html">two</a>
    </li>
    <li>
        <a href="three.html">three</a>
        <a href="athree.html">three</a>
    </li>
</ul>
```

one	
two	
three	

관련 메서드(Plugin Methods)

리스트 새로 고침

```
$("ul").listview("refresh");
```

3장: API

기본값 설정(Configuring defaults)

jQuery Mobile 초기화 이벤트(The mobileinit event)

mobileinit 이벤트

mobileinit 이벤트는 jQuery 파일을 불러오고 jQuery Mobile 파일을 불러오기 전에 이벤트를 등록해야 합니다.

스크립트 작성 위치

```
<script src="jquery.js"> </script>
<script src="custom-scripting.js"> </script>
<script src="jquery-mobile.js"> </script>
```

이벤트 작성

```
$(document).bind("mobileinit", function(e)
{
    // 내용
});
```

구성 옵션(Configurable options)

네임스페이스

ns(string, default:"");

```
$.mobile.ns = "my";
```

위와 같이 변경하면 페이지 작성시 다음과 같이 작성해야 합니다.

```
data-role => data-my-role
```

페이지 자동 초기화

autoInitializePage(Boolean, default:true);

값이 false 로 설정되면 page 가 자동으로 초기화되지 않습니다.

```
$.mobile.autoInitializePage = true;
```

서브페이지 키 변경

subPageUrlKey(String, default:"ui-page");

ui-page 로 설정된 서브페이지 키를 다른 명으로 변경할 수 있습니다.

```
$.mobile.subPageUrlKey = "ui-page";
```

활성화된 페이지 클래스명 변경

activePageClass(String, "ui-page-active");

```
$.mobile.activePageClass = "ui-page-active";
```

Ajax 방식 사용 여부

ajaxEnabled(Boolean, default:true);

```
$.mobile.ajaxEnabled = true;
```

기본 페이지 전환 방식 설정

defaultPageTransition(string, default:"slide");

기본으로 사용할 페이지 전환 방식을 설정합니다. none 으로 설정하는 경우 전환 효과가 발생하지 않습니다.

```
$.mobile.defaultPageTransition = "slide";
```


기본 대화 창 전환 방식 설정

`defaultDialogTransition(string, default:"pop");`

기본으로 사용할 대화 창 전환 방식을 설정합니다. none 으로 설정하는 경우 전환 효과가 발생하지 않습니다.

```
$.mobile.defaultDialogTransition = "pop";
```

스크롤 거리 설정

`minScrollBack(string, default:150);`

스크롤을 시작하는 최소 거리를 설정합니다.

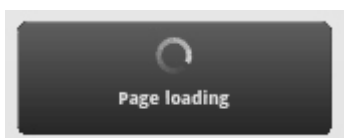
```
$.mobile.minScrollBack = 150;
```

로딩 메시지 설정

`loadingMessage(string, default:"loading");`

로딩시 보여지는 로딩 메시지를 변경합니다.

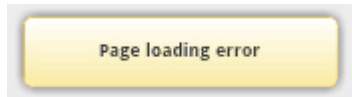
```
$.mobile.loadingMessage = "Page loading";
```



페이지 로드 실패 메세지 설정

pageLoadErrorMessage(string, default:"Error Loading Page");

```
$.mobile.pageLoadErrorMessage = "Error Loading Page";
```



이벤트(Events)

터치 이벤트(Touch Events)

tap

터치 하면 이벤트 발생

```
$("#div[data-role='page']").bind("tap", function (e)
{
    alert("tap");
});
```

taphold

터치한 후 몇 초가 지나면 이벤트 발생

```
$("#div[data-role='page']").bind("taphold", function (e)
{
    alert("taphold");
});
```

swipe

터치한 상태에서 움직이면 이벤트 발생

(1초 안에 세로 20px 이하 범위에서 왼쪽이나 오른쪽으로 30px 이상 움직이는 경우)

```
$("#div[data-role='page']").bind("swipe", function (e)
{
    alert("swipe");
});
```

swipeleft

터치한 상태에서 왼쪽으로 움직이는 경우 이벤트 발생

```
$("#div[data-role='page']").bind("swipeleft", function (e)
{
    alert("swipeleft");
});
```

swiperight

터치한 상태에서 오른쪽으로 움직이는 경우 이벤트 발생

```
$("#div[data-role='page']").bind("swiperight", function (e)
{
    alert("swiperight");
});
```

방향 전환 이벤트(Orientation change events)

방향 전환 이벤트

디바이스의 방향이 가로나 세로로 변경되는 경우 발생하는 이벤트로 파라미터를 통해 orientation 속성으로 디바이스의 상태가 가로인지 세로인지 확인할 수 있습니다. 가로인 경우 portrait 값을 가지고 세로인 경우 landscape 값을 가집니다. (버그로 인해 명칭과 반대인 값이 넘어옵니다.)

```
$("#div[data-role='page']").bind("orientationchange", function (e)
{
    alert(e.orientation);
});
```

스크롤 이벤트(Scroll events)

scrollstart

스크롤이 시작되었을 때 발생하는 이벤트입니다.

```
$("#div[data-role='page']").bind("scrollstart", function (e)
{
    alert("scrollstart");
});
```

scrollstop

스크롤이 종료되었을 때 발생하는 이벤트입니다.

```
$("#div[data-role='page']").bind("scrollstop", function (e)
{
    alert("scrollstop");
});
```

페이지 표시 관련 이벤트(Page show/hide events)

pagebeforeshow

페이지가 화면에 보여지기 직전에 발생하는 이벤트 입니다.
(화면 전환이 시작되기 전에 발생합니다.)

```
$("#div[data-role='page']").bind("pagebeforeshow", function (e)
{
    alert("pagebeforeshow");
});
```

pagebeforehide

페이지가 화면에서 사라지기 직전에 발생하는 이벤트 입니다.
(화면 전환이 시작되기 전에 발생합니다.)

```
$("#div[data-role='page']").bind("pagebeforehide", function (e)
{
    alert("pagebeforehide");
});
```

pageshow

페이지가 화면에 보여진 뒤 발생하는 이벤트 입니다.
(화면 전환이 완료되면 발생합니다.)

```
$("#div[data-role='page']").bind("pageshow", function (e)
{
    alert("pageshow");
});
```

pagehide

페이지가 화면에서 사라지면 발생하는 이벤트 입니다.
(화면 전환이 완료되면 발생합니다.)

```
$("#div[data-role='page']").bind("pagehide", function (e)
{
    alert("pagehide");
});
```

이전 페이지 확인

pageshow 이벤트의 경우 이전 페이지를 확인 할 수 있습니다. 이벤트 핸들러의 두 번째 파라미터 내 prevPage 속성을 통해 확인 가능합니다.

```
$("#div[data-role='page']").bind("pageshow", function (e, ui)
{
    alert(ui.prevPage);
});
```


다음 페이지 확인

pagehide 이벤트의 경우 다음 페이지를 확인 할 수 있습니다. 이벤트 핸들러의 두 번째 파라미터 내 nextPage 속성을 통해 확인 가능합니다.

```
$("#div[data-role='page']").bind("pagehide", function (e, ui)
{
    alert(ui.nextPage);
});
```

페이지 초기화 이벤트(Page initialization events)

pagebeforecreate

페이지가 만들어지기 직전 발생하는 이벤트 입니다.

```
$("#div[data-role='page']").bind("pagebeforecreate", function (e)
{
    alert("pagebeforecreate");
});
```

pagecreate

페이지가 만들어지면 발생하는 이벤트 입니다.

```
$("#div[data-role='page']").bind("pagecreate", function (e)
{
    alert("pagecreate");
});
```

가상 마우스 이벤트(Virtual mouse events)

가상 마우스 이벤트

vmouseover, vmousedown, vmousemove, vmouseup, vclick, vmousecancel

```
$("#div[data-role='page']").bind("vclick", function (e)
{
    alert("vclick");
});
```

애니메이션 이벤트(Animation events)

animationComplete

CSS 애니메이션을 이용해 클래스를 추가하거나 제거했을 때 발생하는 이벤트

```
$("#div[data-role='page']").bind("animationComplete", function (e)
{
    alert("animationComplete");
});
```

메서드 & 유틸리티(Methods & utilities)

화면 변경(changePage)

화면 변경

프로그래밍 방식으로 다른 페이지로 변경하는 방법입니다. 두 가지 파라미터를 사용할 수 있습니다.

to(String or Object) 필수로 들어가야 하는 파라미터 입니다.

options(Object) 사용하지 않아도 됩니다. to 부분에는 URL 값을 넣거나 객체를 넣을 수 있습니다.

```
$.mobile.changePage(to, options);
```

```
$.mobile.changePage("page.html");  
또는  
$.mobile.changePage($("#id");
```

options 부분에는

화면전환(transition(String, default:"slide"))

반대(reverse(Boolean, default:false))

해쉬 업데이트(changeHash(Boolean, default:true))

역활(role(String, default:undefined))

페이지 컨테이너(pageContainer(jQuery Collection, default:\$.mobile.pageContainer))

방식(type(String, default:"get"))

데이터(data(Object or String, default:undefined))

페이지가 있는 경우 새로고침 여부(reloadPage(Boolean, default:false)

가 들어갈 수 있습니다.

```
$.mobile.changePage("page.html", {transition:"flip"})
```

페이지 로드(loadPage)

페이지 로드

외부 페이지를 로드만 하는 방법입니다. `changePage()` 메서드와 같이 페이지 전환은 되지 않습니다.

`$.mobile.loadPage(to, options);`

options 부분에는

역할(role(String, default:undefined))

페이지 컨테이너(pageContainer(jQuery Collection, default:\$.mobile.pageContainer))

방식(type(String, default:"get"))

데이터(data(Object or String, default:undefined))

페이지가 있는 경우 새로고침 여부(reloadPage(Boolean, default:false)

가 들어갈 수 있습니다.

로딩메세지 보여주기(showPageLoadingMsg)

로딩메세지 보여주기

임의로 로딩메세지를 보여주는 메서드입니다.

```
$.mobile.showPageLoadingMsg();
```

로딩메세지 숨기기(hidePageLoadingMsg)

임의로 로딩메세지를 숨겨주는 메서드입니다.

```
$.mobile.hidePageLoadingMsg();
```

URL 정보(parseUrl)

URL 정보

URL 에 대한 정보 분석을 위한 메서드 입니다.

```
var info = $.mobile.path.parseUrl("info.html");
```

반환되는 정보는 hash, host, hostname, href, pathname, port, protocol, search, authority, directory, domain, filename, hrefNoHash, hrefNoSearch, password, username 이 있습니다.

절대경로 생성(makePathAbsolute)

절대경로 생성

두 경로를 비교하여 절대경로를 문자열을 반환해줍니다.

```
var path = $.mobile.path.makePathAbsolute(relPath, absPath);
```

```
// return : /a/b/c/file.html
var path = $.mobile.path.makePathAbsolute("/a/b/c/bar.html", "file.html");

// return : /a/foo/file.html
var path = $.mobile.path.makePathAbsolute("../foo/file.html", "/a/b/c/bar.html");
```

도메인 내에서의 절대경로 생성(makeUrlAbsolute)

도메인 내에서의 절대경로 생성

도메인을 포함해 상대경로를 절대경로로 변환해줍니다.

```
var path = $.mobile.path.makeUrlAbsolute(relPath, absPath);
```

```
// return : http://foo.com/a/b/c/file.html
var path = $.mobile.path.makeUrlAbsolute("file.html ", "http://foo.com/a/b/c/test.html");

// return : http://foo.com/a/foo/file.html
var path = $.mobile.path.makeUrlAbsolute("../foo/file.html", "http://foo.com/a/b/c/test.html");

// return : http://foo.com/bar/file.html
var path = $.mobile.path.makeUrlAbsolute("/foo.com/bar/file.html", "http://foo.com/a/b/c/test.html");

// return : http://foo.com/a/b/c/test.html?a=1&b=2
var path = $.mobile.path.makeUrlAbsolute("?a=1&b=2", "http://foo.com/a/b/c/test.html");

// return : http://foo.com/a/b/c/test.html#bar
var path = $.mobile.path.makeUrlAbsolute("#bar", "http://foo.com/a/b/c/test.html");
```


같은 도메인 여부 확인(isAbsoluteUrl)

같은 도메인 여부 확인

URL 의 도메인을 비교하기 위한 메서드 입니다.

```
var path = $.mobile.path.isSameDomain(url1, url2);
```

```
// return : true
var path = $.mobile.path.isSameDomain(http://foo.com/a/file.html, "http://foo.com/a/b/c/test.html");

// return : false
var path = $.mobile.path.isSameDomain(file://foo.com/a/file.html, "http://foo.com/a/b/c/test.html");

// return : false
var path = $.mobile.path.isSameDomain(https://foo.com/a/file.html, "http://foo.com/a/b/c/test.html");

// return : false
var path = $.mobile.path.isSameDomain(http://foo.com/a/file.html, "http://bar.com/a/b/c/test.html");
```

상대경로 여부 확인(isRelativeUrl)

상대경로 여부 확인

URL 이 상대경로인지 여부를 확인하기 위한 메서드 입니다.

```
var path = $.mobile.path.isRelativeUrl(url);
```

```
// return : false
var path = $.mobile.path.isRelativeUrl("http://foo.com/a/file.html");

// return : true
var path = $.mobile.path.isRelativeUrl("//foo.com/a/file.html");

// return : true
var path = $.mobile.path.isRelativeUrl("/a/file.html");

// return : true
var path = $.mobile.path.isRelativeUrl("file.html");

// return : true
var path = $.mobile.path.isRelativeUrl("?a=1&b=2");

// return : true
var path = $.mobile.path.isRelativeUrl("#foo");
```

절대경로 여부 확인(isAbsolutePath)

절대경로 여부 확인

URL 이 절대경로인지 여부를 확인하기 위한 메서드 입니다.

```
var path = $.mobile.path.isAbsolutePath(url);
```

```
// return : true
var path = $.mobile.path.isAbsolutePath("http://foo.com/a/file.html");

// return : false
var path = $.mobile.path.isAbsolutePath("//foo.com/a/file.html");

// return : false
var path = $.mobile.path.isAbsolutePath("/a/file.html");

// return : false
var path = $.mobile.path.isAbsolutePath("file.html");

// return : false
var path = $.mobile.path.isAbsolutePath("?a=1&b=2");

// return : false
var path = $.mobile.path.isAbsolutePath("#foo");
```

스크롤 이동(silentScroll)

스크롤 이동

스크롤 관련 이벤트를 발생시키지 않고 특정 위치로 스크롤하는 메서드 입니다.

```
$.mobile.silentScroll(100);
```

해상도 최대 최소값 설정(addResolutionBreakpoints)

해상도 최대 최소값 설정

HTML 요소에 클래스가 적용될 폭을 설정합니다.

```
// 400px 에 중단점 추가
$.mobile.addResolutionBreakpoints(400);

// 2 개 이상의 중단점 추가
$.mobile.addResolutionBreakpoints([600, 800]);
```

활성화된 페이지 찾기(activePage)

활성화된 페이지 찾기

현재 페이지를 참조하는 프로퍼티 입니다.

```
$.mobile.activePage;
```

반응하는 레이아웃(Responsive Layout)

방향에 따른 클래스(Orientation Classes)

방향에 따른 클래스

디바이스의 가로/세로 방향에 따른 CSS 를 활용할 수 있습니다.(버그로 인해 명칭과 반대되는 값이 적용됩니다.)

가로방향 CSS

```
.portrait {  
    /* 가로방향일 때 CSS */  
}
```

세로방향 CSS

```
.landscape {  
    /* 가로방향일 때 CSS */  
}
```

크기에 따른 클래스(Min/Max Width Breakpoint Classes)

크기에 따른 클래스

디바이스의 너비에 따라서 설정되는 클래스 입니다. 너비는 320, 480, 768, 1024px 점을 가집니다.

일반 클래스

```
.myelement {  
    /* 일반 클래스 */  
}
```

너비가 480px 이하일 때 적용되는 클래스

```
.min-width-480px .myelement {  
    /* 너비가 480px 이하일 때 적용되는 클래스 */  
}
```

크기 제어 포인트(Adding Width Breakpoints)

크기 제어 포인트

크기에 따른 클래스 설정시 기존 포인트 외에 새로운 포인트를 추가하고 싶을 경우 사용합니다.

```
// 400px 에 중단점 추가
$.mobile.addResolutionBreakpoints(400);

// 2 개 이상의 중단점 추가
$.mobile.addResolutionBreakpoints([600, 800]);
```

미디어 쿼리(Running Media Queries)

미디어 쿼리 실행 가능 여부 테스트

CSS 의 특정 미디어 쿼리 적용 여부를 테스트하는 기능입니다. 지원하면 true, 지원하지 않으면 false 가 반환됩니다.

```
// screen 미디어 형식 테스트
$.mobile.media("screen");

// screen 미디어 형식과 min-width 테스트
$.mobile.media("screen and (min-width: 480px)");

// screen 미디어 형식과 레티나 디스플레이 여부 테스트
$.mobile.media("screen and (-webkit-min-device-pixel-ratio: 2)");
```