

목 차

I	Overview	1
II	표준화면 정의	2
1	개요 및 원칙	2
2	디렉터리 구성 정의	3
3	디렉터리 구성 정의	4
4	디자인 원칙 및 제약사항	5
III	웹 인터페이스 디자인 가이드	5
1	디자인 가이드 정의	6
2	디자인 구성 요소	11
3	HTML5와 CSS3.0 기본활용	22
IV	환경 구성	26
1	지원 플랫폼	26
2	지원 사양	27
V	사용자 화면 구성 지원 API	29
1	Configuring Defaults	29
2	Events	31
3	Methods & Utilities	34
4	Responsible Layout	42
5	Dynamic Page Generation	44
VI	UX Component	48
1	Button	48
2	Panel	50
3	Internal / External Link	52
4	Label / Text	55
5	Tabs	57
6	Form	61
7	Menu	66
8	Progress Dialog / Bar	68
9	Dialog	69
10	Grid View	72
11	Table / List	75
12	Date / Time Picker	77
13	Check Box	80
14	Radio Button	82
15	Icon	84

16	Selector/Switch	87
17	Collapsible Block	92
VII	페이지 구조타입	94
1	모바일 Page	94
2	Header	97
3	Footer	100
4	Theme Type	103

I 사용자 경험(UX) 지원 기능 소개

전자정부 모바일 표준프레임워크의 **사용자 경험(UX) 지원 기능**은 **스마트 기반 모바일 환경에서 최적화된** 화면 처리 기능, 개발도구, 모바일 특화 컴포넌트 제공을 통해 모바일 웹 서비스개발에 대한 생산성 향상, 모바일 전자정부 시스템 간의 컴포넌트 재사용성 향상, 연계 표준을 통한 상호 운용성 향상, 응용 소프트웨어의 표준화 효과를 제공합니다.

사용자 경험(UX)지원 기능의 특징

- ▶ 스마트 기반의 다양한 모바일 디바이스 플랫폼에 통합된 사용자 인터페이스를 제공합니다.
- ▶ 스마트폰의 터치 이벤트 기반에 가장 최적화된 전자정부 모바일 표준 사용자 경험(UX/UI)을 제공합니다.
- ▶ HTML5, CSS3, JavaScript의 경량화된 코드로 조합되어 있는 구조는 터치 이벤트 환경에 맞는 화면 구성을 제공합니다.
- ▶ 유연하고 쉽게, 모바일에 특화 된 화면의 그래픽 레이아웃을 설계·적용하고, 다양한 테마를 디자인 할 수 있습니다.
- ▶ 복잡한 cross-platform & cross-device 환경에서 터치 이벤트에 최적화된 그래픽 레이아웃을 제공 합니다.
- ▶ 개발 편의성을 고려하여 쉽고 편하게 개발을 진행하고 적용할 수 있도록 개발자 표준 코드 및 예제가 적용된 개발자 가이드 기능을 제공합니다.

전자정부 모바일 표준프레임워크의 사용자 경험(UX)지원 기능 제공 전략

Cross-platform & Cross-device를 통한 Any Device Any platform Any OS 서비스를 지향 합니다.



I 개요 및 원칙

개요

전자정부 모바일 표준프레임워크는 모바일 웹 어플리케이션 구축 시 화면 구현이 필요한 기능에 대해 기본적인 화면 표준을 정의하여 개발 및 유지보수를 위한 화면 개발 가이드를 제시한다.

표준정의 원칙

본 문서는 모바일 공통 컴포넌트 및 개발프레임워크 구축 사업 범위 내에 있는 모바일 웹 어플리케이션의 테스트 및 향후 전자정부 사업에 적용을 위한 가이드 제시를 목적으로 작성되며, 따라서 일반적인 모바일 웹 어플리케이션 개발 사업에서 작성되는 화면 표준정의서 중 반드시 필요한 부분에 대해서만 언급한다.

I 디렉터리 구성 정의

웹 디자인 이미지 디렉터리의 명명 규칙 이외의 사항은 별도의 "관리환경 개발가이드"에 의하여 규정한다. 웹 디자인 이미지 디렉터리 명명은 아래와 같다. (본 가이드라인은 하나의 예시로서 사용자의 개발환경에 따라 충분히 변경 가능함)

Web-root/images/표준코드/이미지파일명

예) /oe1/cms/com/confirm.gif

→ **이미지 파일명**

→ **표준코드**-[시스템 코드]-[서브시스템 코드]-[구분자]

I 파일 명명규칙 정의 (Naming Rule)

웹 디자인 이미지 파일의 명명 규칙 이외의 사항은 별도의 "운영환경 개발가이드"에 의하여 규정한다.

이미지타입 구분코드

이미지의 사용 용도에 따라 아래와 같이 유형을 구분하여 파일 명명에 사용한다.

이미지 유형	형태
bt	버튼 이미지
bul	글머리표
img	이미지
img_event	이벤트 페이지용 이미지
txt	Text 형 이미지
title	타이틀용 이미지
title_s	소 타이틀용 이미지
tttitle	테이블 타이틀용 이미지
tab	Tab Menu 용 이미지
topmn_	유틸리티 메뉴용 이미지(상단메인)
bttmn_	Bottom 용 이미지 (하단)
ban_	Banner 형 이미지

※ 위 유형은 화면 구성 시 추가 혹은 변경될 수 있음.

파일 명명규칙

파일 이름은 소문자로 작성하며, 서브시스템 코드와 유형 일련번호의 조합으로 구성한다. (본 가이드라인은 하나의 예시로써 사용자의 개발환경에 따라 충분히 변경 가능함)

[서브시스템 코드]_[이미지 유형]_일련번호

예) bbs_title_001

└─> 일련번호

└─> 이미지 유형 - 타이틀용 이미지

└─> 서브시스템 **BBS** - 커뮤니티 > 게시판

I 디자인 원칙

이미지는 로딩>Loading>이 신속하게 진행될 수 있도록 제작해야 한다.

- ▶ 시스템의 심미성 향상을 위해 사용되는 이미지와 그래픽 불릿>Bullet>, 아이콘 등의 시각적 요소들은 페이지의 복잡도를 줄이기 위해 사용을 최소화해야 한다. 페이지의 로딩 속도에 영향을 미치지 않고 효과를 극대화시키기 위해서는 이미지의 사이즈를 작게 설정하고 컬러의 수를 줄여서 사용해야 한다.

이미지의 표현방식은 일관되게 제시해야 한다.

- ▶ 시스템 전반에 걸쳐서 사용자에게 통일된 이미지의 개념을 보여주고, 시스템에 대한 일관된 이미지를 전달 할 수 있어야 한다.

시스템이 기본적으로 제공하는 폰트 타입을 사용해야 한다.

- ▶ 폰트의 스타일은 가능하면 시스템에서 제공하는 기본 폰트 스타일을 사용하여 특정 폰트 지원여부에 영향을 받지 않도록 해야 한다. 폰트의 사용은 가독성>Readability>을 우선적으로 고려해야 하며, 시각적으로 명확하지 않은 폰트를 사용하여 텍스트를 읽을 때 불편함이 생기지 않도록 해야한다.

폰트는 정보의 분류 체계, 중요도, 목적에 따라 차별되게 적용되어야 한다.

- ▶ Bold 체와 Italic 체는 중요도가 높거나 구분될 필요가 있는 경우에만 사용해야 하며, 지나치게 자주 사용하면 정보의 가독성을 저해할 수 있다.
- ▶ 폰트 하단의 밑줄은 링크와 혼동될 수 있으므로 가급적 사용을 자제한다.

동일한 페이지에서 폰트 스타일과 크기는 3종류 이내로 제한하여 사용한다.

- ▶ 한 페이지 안에 3종류 이상의 폰트 스타일을 사용하면 일관성이 저해되어 가독성이 떨어진다. 따라서 시각적 통일성을 유지하는 범위 내에서 사용해야 한다.

한 페이지 내에 표현할 수 있는 최대 컬러 군은 5가지 이다.

- ▶ 5가지가 넘는 컬러 군을 사용할 경우 페이지 구성을 혼란하게 하여 수행 능력을 저해할 수 있다.

컬러는 시스템의 고유 아이덴티티>Identity>가 적용될 수 있어야 한다.

- ▶ 사용자들에게 대상 시스템이나 제품에 대한 공통적인 브랜드 이미지를 심어 줄 수 있도록 시스템의 특성이 반영된 컬러>CI>를 적극적으로 적용해야 한다.

아이콘은 텍스트와 결합하여 제시해야 한다.

- ▶ 아이콘은 시각적 효과가 높지만 정확한 메타포>metaphor>를 표현하기가 어려우므로 텍스트와 결합하여 정확도를 높여야 한다.

동일한 모양의 아이콘에 중복된 기능을 부여하지 않는다.

- ▶ 하나 이상의 기능을 동일한 모양의 아이콘에 부여하지 말아야 한다.

아이콘은 일정한 위치에 배치해야 한다.

- ▶ 아이콘을 페이지 별로 상이한 곳에 배치하면 사용자가 아이콘을 직관적으로 이해하는데 방해가 되어 아이콘 이용에 어려움을 줄 수 있다.

I 디자인 가이드 정의

전자정부 모바일 표준프레임워크는 기본원칙, 사용자 및 환경정의, 페이지 레이아웃 정의에 대한 디자인 가이드를 제공한다.

기본 원칙

- ▶ 사용자 경험에 초점을 맞춘다.
- ▶ 소수의 핵심 기능에 집중한다.
- ▶ 디바이스 호환성, 확장성을 고려한다.
- ▶ 정보 소외계층의 접근성을 고려한다.
- ▶ 웹 표준을 준수하여 개발한다.
- ▶ 일관성을 유지한다.
- ▶ 이미지를 제한적으로 사용한다.
- ▶ 모바일 환경의 유해 요소를 제거한다
- ▶ 웹 네비게이션을 최적화한다.

사용자 및 환경 정의

사용자

- ▶ 대상 사용자 군에 따라 공공 모바일 서비스는 크게 대국민 서비스와 행정 서비스, 두 가지로 분류될 수 있다.
- ▶ 대국민 서비스는 모든 대한민국 국민을 대상으로 한다. 이는 다양한 연령, 성별 및 지적·사회적 수준까지 고려해야 함을 의미한다. 그러나 개별 서비스는 그 성격과 목적에 따라 국민 중 특별히 정의된 사용자 군을 대상으로 할 수 있다.
- ▶ 행정 서비스는 해당 서비스에 관련된 공무원을 대상으로 한다.

모바일 환경

- ▶ 대국민 모바일 서비스는 사용자의 다양한 디바이스 환경에서 호환 가능해야 한다.
- ▶ 모바일 웹은 HTML, CSS, Javascript를 사용하여 구현된다.
- ▶ 디바이스는 풀터치로 사용하는 스마트폰과 태블릿 PC를 대상으로 하며, 피쳐폰은 고려하지 않는다. 디바이스는 최신 버전뿐만 아니라 하위버전의 기기도 고려한다.
- ▶ 모바일 디바이스 브라우저의 기본 해상도는 320X480px 이며 고해상도 스마트폰의 지속적인 출시에 따라 확장성을 고려해야 한다. 또한 스마트폰의 PPI(Pixel Per Inch)는 LCD모니터의 PPI(72ppi)보다 높기 때문에 디자인 작업 시 PC환경 보다 실제 적용 대상 디바이스에서 확인해야 한다.
- ▶ 모바일 환경은 탭(tap) 이벤트를 통해 동작하기 때문에 PC 웹과의 차이점에 대한 충분한 이해를 기반으로 설계해야 한다. 예를 들어, 모바일 화면에서는 페이지 내에서의 스크롤바 사용이 불가능하기 때문에 웹에서 사용하는 프레임 형식을 사용할 수 없다. 또한 터치스크린 기반의 디바이스는 마우스 롤오버(mouse rollover) 같은 이벤트를 사용할 수 없다.
- ▶ 스마트폰과 태블릿은 UI의 기본원칙은 동일하나 화면의 해상도 및 레이아웃에서는 차이를 보인다. 현재 태블릿 PC의 기준 해상도는 600*1024px, 768*1024px 이다.
- ▶ 태블릿은 일반적으로 모바일 웹이 아닌 일반 웹사이트를 접속하여 브라우징 하는 경우가 대부분이나, 웹사이트에서 많이 사용되는 플래시, GIF 애니메이션, 동영상 등의 몇몇 기술은 특정 태블릿에서 동작하지 않는다. 따라서 모바일 웹 뿐만 아니라 태블릿 환경을 고려한 일반 웹사이트를 구축하기 위해서는 본 가이드에서 제시하는 기본 원칙을 준수하여 개발할 것을 권장한다.

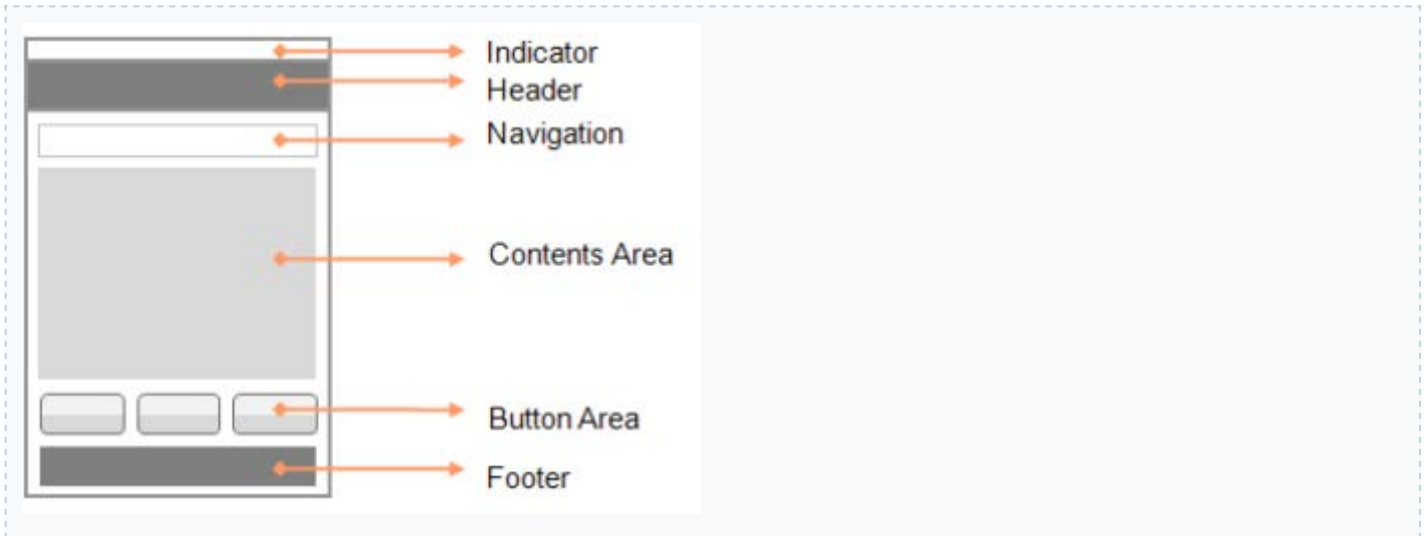
페이지 Layout 정의

구성요소

일반적으로 화면을 구성하는 요소들은 다음과 같으며, 이러한 요소들을 조합하여 다양한 화면을 구성할 수 있다.

- ▶ Indicator 영역 : 각종 서비스의 알림 및 수신, 네트워크, 배터리 상태 등을 아이콘화 하여 표시한다.
- ▶ Header 영역 : 기관의 로고 또는 사이트 명을 표시하여 사이트의 아이덴티티를 전달한다.
- ▶ Navigation 영역 : 주로 홈이나 이전 페이지 또는 최상위 메뉴와 같은 주요 링크만 제공하며 모바일웹의 경우 브라우저 내비게이션 버튼을 활용하므로 생략하는 경우도 있다.
- ▶ Content 영역 : 사용자에게 전달하고자 하는 주요 정보를 담는다.

- ▶ Button 영역 : 필요한 경우에만 사용하며, 버튼은 해당 콘텐츠 아래에 배치하는 것이 일반적이나 상단에 위치시키는 경우도 있다.
- ▶ Footer 영역 : 저작권, 연락처 PC 웹사이트로의 링크 등의 정보가 표시된다.



기본 원칙

- ▶ 대국민 서비스의 Header 영역에는 공공 서비스로서의 Identity 를 드러낼 수 있는 사이트명, 로고, 혹은 소속 기관명이 노출되어야 한다.
- ▶ 대국민 서비스의 Footer 영역에는 PC 버전의 웹 사이트가 있는 서비스의 경우 PC 버전으로의 링크를 제공하는 것이 좋다. (행정 서비스는 예외)
- ▶ 모바일 서비스가 제공되는 웹 사이트의 경우에는 하단에 모바일 버전으로의 링크가 제공되도록 가이드 하는 것이 좋다.
- ▶ Footer 영역은 좁은 공간을 고려하여 PC 버전의 웹 사이트 보다는 간략화 된 내용을 담도록 한다.
- ▶ 유형에 따라 레이아웃을 일관되게 적용하여 시각적인 안정감을 주는 것이 좋다.
- ▶ 콘텐츠의 정렬이 적절히 이루어지도록 그리드 시스템을 사용하는 것이 좋다.
- ▶ 공공 모바일 서비스에서 적합한 레이아웃은 콘텐츠를 읽기 쉽고 보기 좋게 효율적으로 배치하는 것이다. 사용자가 원하는 정보를 쉽게 얻을 수 있도록 사용 환경과 방문 목적을 충분히 고려하여 정보를 배치해야 한다.
- ▶ 콘텐츠의 효과적인 전달을 위해 사용자가 반드시 읽어야 하는 주요 콘텐츠는 주목도가 높은 곳에 우선 배치하고 관심을 끌 수 있는 스타일을 적용하는 것을 권장한다.
- ▶ 여백을 적절히 사용하여 시선의 흐름을 유도하는 것이 좋다.

모바일 레이아웃

■ 작은 스크린에 적합한 콘텐츠

- ▶ 대국민 서비스를 모바일 버전으로 전환할 때 PC 버전의 웹 사이트에 있는 모든 정보를 모바일 버전으로 전환하는 것은 디바이스 화면 크기의 물리적 제약으로 인해 무척 어렵기 때문에 주요 콘텐츠에 대한 선택적으로 적용해야 한다.
- ▶ 동일한 콘텐츠를 PC 화면과 모바일 화면에 담았을 때, 모바일의 경우에는 PC 에 비해 사용자의 이해도가 현저하게 떨어진다는 점을 고려하여 상대적으로 큰 글씨와 간결한 내용을 담도록 한다.
- ▶ 작은 화면에 너무 많은 콘텐츠를 배치하면 사용자가 원하는 콘텐츠에 빠르게 접근하기 어렵다. 긴 내용은 여러 페이지로 적절하게 나누어 배치하는 것이 좋다.

■ 세로형 배치

모바일 화면은 가로 스크롤보다는 세로 스크롤이 편리하게 사용되므로 공공서비스의 콘텐츠 레이아웃은 단일 칼럼의 세로 레이아웃을 기본형으로 적용한다. 그러나 서비스의 성격에 따라 가로 레이아웃을 기본형으로 적용하는 것이 가능하다.



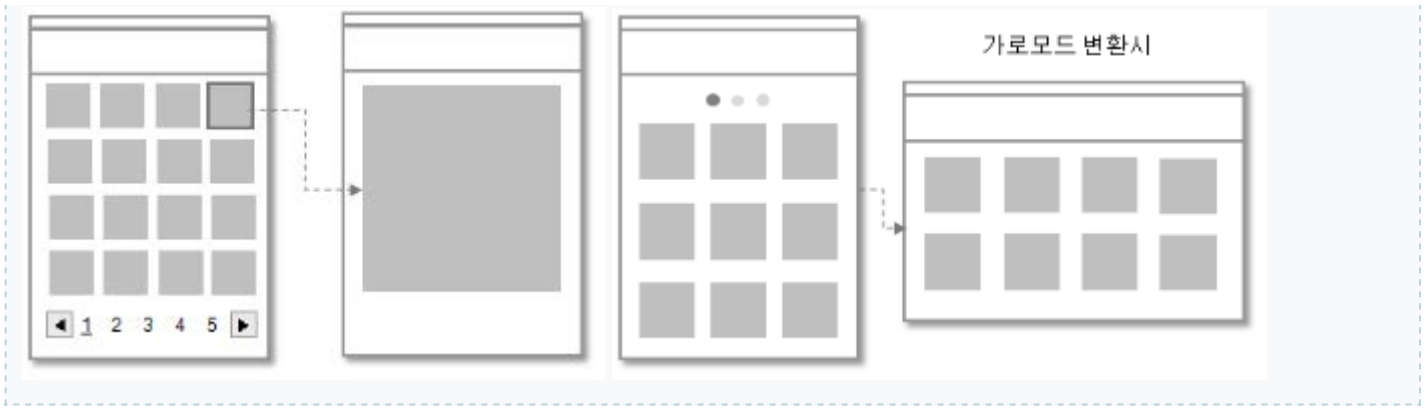
■ 기본형태

- ▶ 리스트형 레이아웃은 정보성 데이터가 많은 대국민/행정 서비스에서 적합한 레이아웃이다.
- ▶ 썸네일 뷰보다는 긴 텍스트 타이틀을 보여줄 수 있는 장점이 있으며 비교적 얇은 계층구조의 표현이 가능하다.
- ▶ 동일 레벨의 내용을 일정한 기준에 따라 순차적으로 정렬한다.
- ▶ 한 화면에 목록을 다 보여줄 수 없는 경우, 하단에 마지막 목록의 일부분만 나타내어 목록이 계속 있다는 시각적인 단서를 제공한다. ①
- ▶ 목록의 상하 구조를 표현하는 경우 시각적인 구분을 위하여 아이콘이나 블릿을 사용하기도 한다. ②
- ▶ 모든 목록을 한꺼번에 가져오기 어려운 경우, 우선 일부 목록만 제공하고 사용자가 필요한 경우 추가 목록을 불러올 수 있도록 한다. ③
- ▶ 편집모드에서는 목록 앞에 체크 박스를 두거나 목록 뒤에 삭제 버튼 등을 두어 기본 목록 화면과는 명확하게 구분되도록 한다. ④



■ 이미지형 List (Gallery)

- ▶ 리스트를 이미지형식으로 표현하는 레이아웃 스타일이다.
- ▶ 사진 리스트와 같은 이미지 데이터를 보여주거나 메인 화면의 메뉴선택과 같은 경우 적용 가능한 유형이다.
- ▶ 이미지 개수가 많아지는 경우 하단스크롤 혹은 페이지 내비게이션이 제공된다.
- ▶ 메뉴를 표현하기 위한 형태로 사용되는 경우 메뉴의 개수가 지나치게 많아져서 하단으로 길게 스크롤 되는 것은 바람직하지 않다.
- ▶ 이미지 형식의 메뉴이므로 메뉴 별로 개별적인 디자인 작업이 필요하다.
- ▶ 이미지크기에 따라 배치되는 이미지의 개수가 달라질 수 있으며 중앙정렬의 등간격 배치가 되도록 한다.
- ▶ 상세화면에서는 이미지를 확대/축소하는 기능이 지원된다.
- ▶ 가로모드로 전환할 때는 우측 빈 공간이 생기지 않도록 가로 폭에 보여지는 메뉴 개수를 더 늘려서 배치하는 것이 좋다.



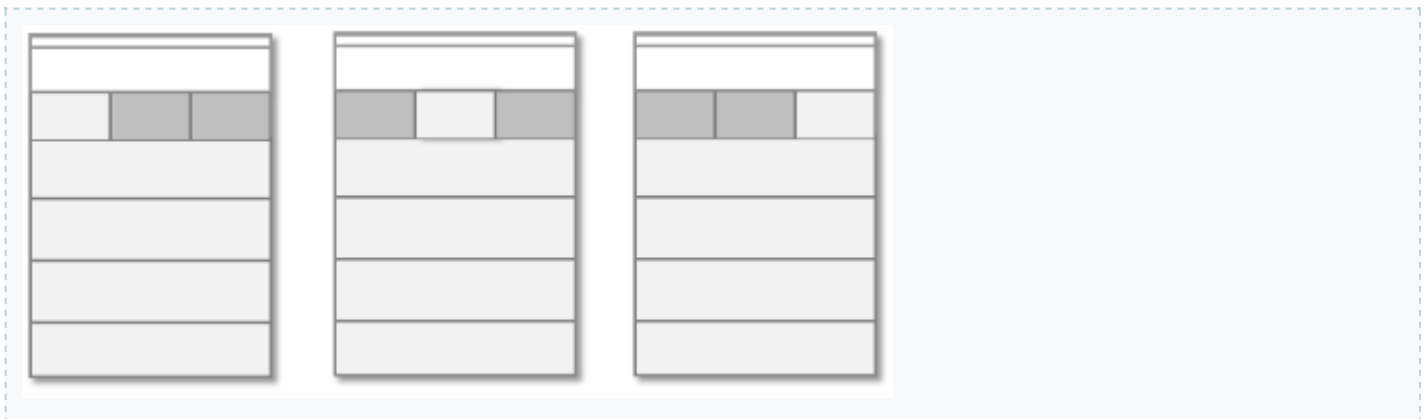
■ Indexed List

- ▶ 리스트 형태의 레이아웃에 인덱스(Index)가 제공되는 형태이다.
- ▶ 주소록처럼 한 화면에 제공되는 목록이 많아서 시각적으로 그룹핑이 필요할 경우 사용되며, 일반 리스트형태에 비해 시각적으로 원하는 정보로의 검색이 용이한 장점이 있다.
- ▶ 정보량이 많은 행정 서비스 등에서 적용 가능한 유형의 레이아웃이다.
- ▶ 인덱스 그룹은 서로 중복되거나 종속되지 않고 명확하게 구분할 수 있어야 한다.
- ▶ 깊은 계층구조의 메뉴를 보여주는 유형에는 적합하지 않다.



■ Tabbed List

- ▶ 탭 버튼 아래 목록이 같이 제공되는 화면으로 하나의 화면에서 여러 개의 목록을 한번에 보여줄 수 있다.
- ▶ 세부 카테고리가 나뉘어있는 목록 콘텐츠에 적합하며, 탭 명은 목록에 대한 타이틀 역할을 한다.
- ▶ 탭 버튼을 선택하면 하단에 해당 목록의 데이터가 나오며, 일반적으로 첫 번째 탭의 목록이 열려있는 상태이다.
- ▶ 비교적 많은 양의 정보를 한 화면에서 내비게이션할 수 있기 때문에 정보전달성 서비스에서 가장 많이 사용되는 유형이다.

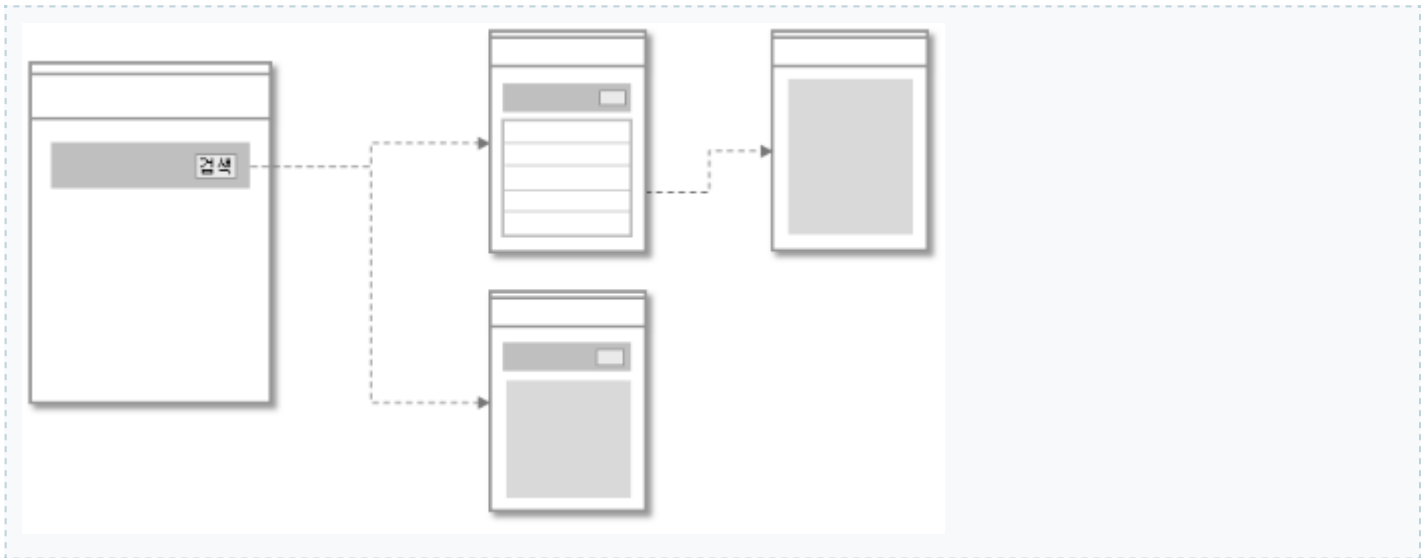


Search

■ 기본 원칙

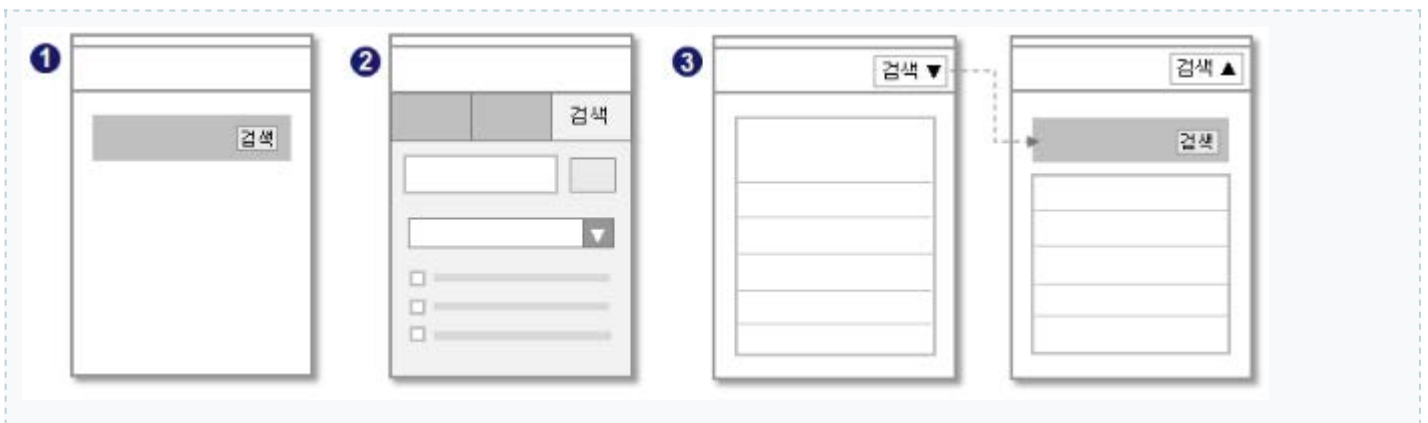
- ▶ 검색은 많은 정보를 담고 있는 대국민/행정 서비스에서 사용자가 빠르게 원하는 서비스에 접근할 수 있도록 해주기 위해 제공하는 기능이다.
- ▶ 검색창의 위치는 화면 상단에 위치하는 경우가 많으며, 경우에 따라 화면의 상 하단에 중복 노출시키는 경우도 있다.
- ▶ 검색어를 입력하고 버튼을 누르면 그에 대한 결과 값을 보여준다.

- ▶ 검색에 대한 결과 건수가 1건 일 경우 해당 결과에 대한 상세 화면이 제공되고, 검색에 대한 결과 건수가 2건 이상 일 경우 목록 화면 그리고 상세화면의 순으로 제공된다.
- ▶ 포탈 성격의 대국민 서비스의 경우 상단에 검색기능을 제공해 주는 것이 권장된다.



■ 검색 제공 방식

- ▶ 검색 활용 빈도가 높은 경우 주요 화면 내에 검색창을 제공한다. ①
- ▶ 검색 옵션이 다양하여 사용자 설정이 많은 경우 검색 화면을 별도로 제공한다. ②
- ▶ 다른 콘텐츠의 공간 확보를 위하여, 검색이 필요한 경우에만 사용자가 별도로 검색창을 꺼내서 사용할 수 있도록 한다. ③



■ 텍스트 검색

- ▶ 검색어를 텍스트로 입력할 수 있는 입력 필드와 검색버튼이 제공된다.
- ▶ 검색창은 주로 상단에 제공되거나 텍스트를 입력하기 위한 키패드가 생성되면 자동적으로 상단으로 올라간다.
- ▶ 사용자의 편의를 고려하여 초성검색이나 자동완성기능과 같은 검색어를 제안하는 방식이 적용되기도 한다.



I 디자인 구성요소

전자정부 모바일 표준프레임워크는 화면 내비게이션, 화면 구성요소 정의, 해상도의 정의에 대한 디자인 가이드를 제공한다.

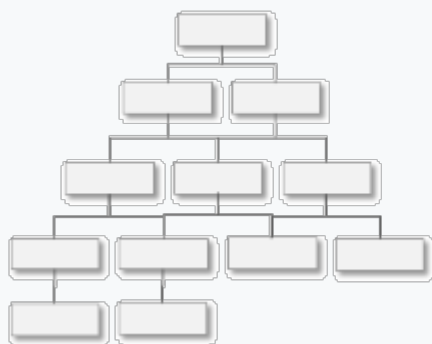
화면 Navigation

기본 원칙

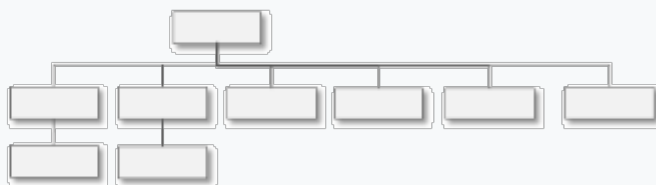
▶ 정보 구조의 단순화

정보 계층의 폭(width)은 계층의 최상위 단계에서 제공하는 옵션들의 수를 의미하며, 깊이(depth)는 계층의 단계 수이다. 정보의 계층이 너무 깊으면 다른 정보를 찾기 위한 이동방법이 복잡해지기 쉽다. 따라서 모바일 환경에서는 3 depth 이상 깊이는 지양하고 화면 연결 설계 시 정보 계층의 폭과 깊이를 고려하여 콘텐츠에 빠르게 접근할 수 있고 콘텐츠 추가도 용이하도록 구성해야 한다.

좁고 깊은 구조 (x)



넓고 얕은 구조 (O)

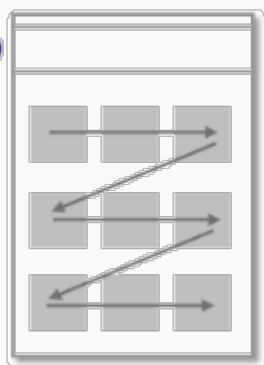


메뉴

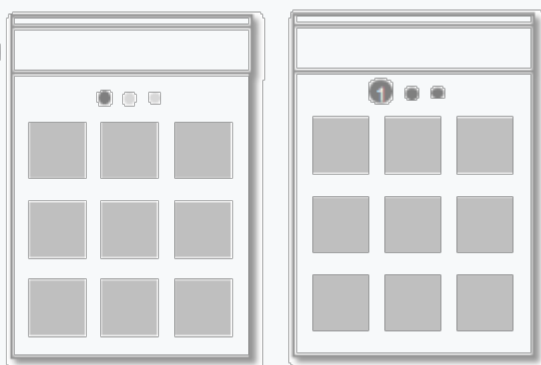
■ Grid 형식 메뉴

- ▶ 비슷한 레벨의 콘텐츠 여러 개를 아이콘화하여 배치할 경우 사용된다.
- ▶ 메뉴 정렬 순서는 좌측에서 우측, 위에서 아래 방향으로 배치한다. ①
- ▶ 2개 이상의 메뉴 페이지가 존재하는 경우 현재 메뉴 페이지의 위치와 전체 메뉴 페이지 개수에 대한 시각적인 단서를 제공한다. ②

①

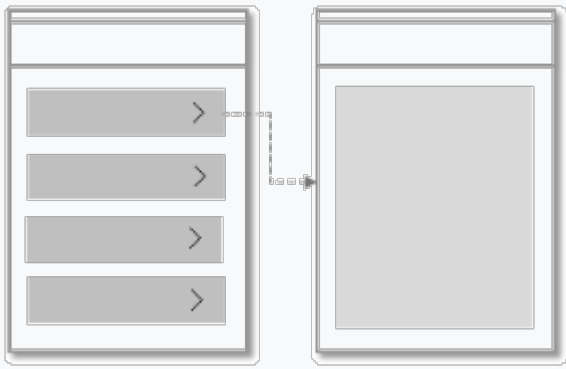


②



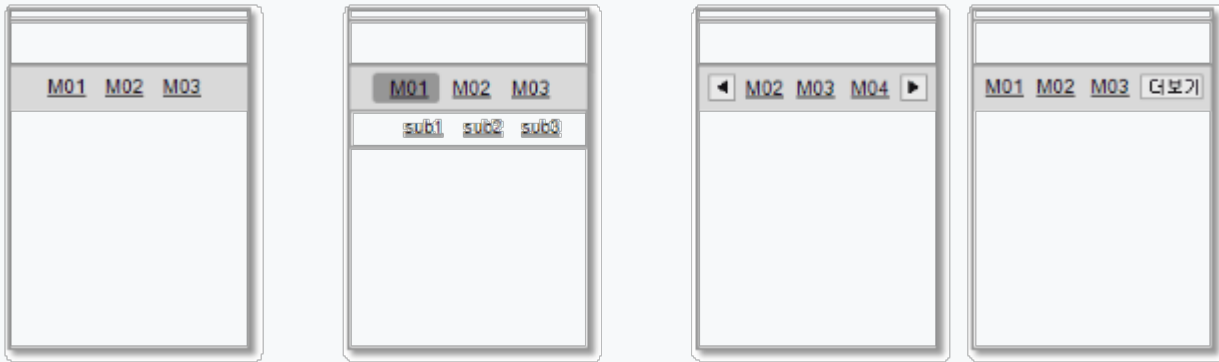
■ List 형식 메뉴

- ▶ 각 메뉴마다 하위 콘텐츠를 대표하는 성격의 레이블이나 아이콘을 제공한다. 추가 메뉴에 대한 공간 제약이 덜하므로 확장성이 좋다.



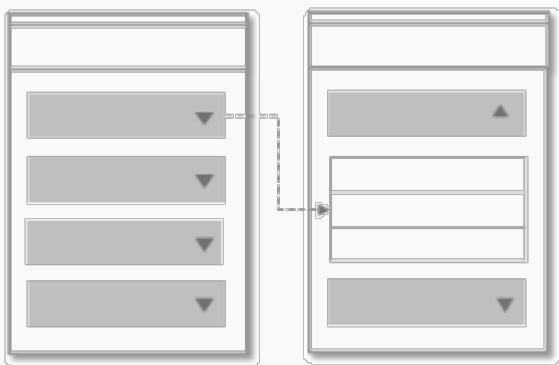
■ Bar 형식 메뉴

- ▶ 텍스트 형식의 메뉴표현으로 웹화면에서의 Global Navigation Bar와 유사한 형태의 내비게이션 바이다.
- ▶ 메뉴명을 가로로 배치한다.
- ▶ 모든 화면마다 고정된 메뉴영역을 제공해야 할 때 사용된다.
- ▶ 한 화면에서 레벨이 다른 메뉴간 이동이 가능하다.
- ▶ 콘텐츠 영역 확보를 위한 공간의 제약 때문에 2 depth 이하의 정보 구조를 적용하는 것이 적합하다.
- ▶ 화면에 모든 메뉴명이 표현되지 못하였을 경우 숨겨져 있는 메뉴를 볼 수 있는 시각적인 단서를 제공한다.



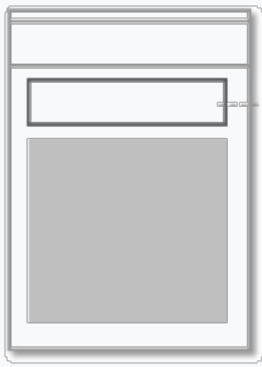
■ Accordion 형식 메뉴

- ▶ 텍스트 형식의 메뉴표현으로 메뉴 목록을 접고 펼치는 방식을 적용한다.
- ▶ 화면에서 메뉴가 차지하는 영역이 크므로 보통 첫 화면에만 제공되는 편이다.
- ▶ 하나의 화면에서 레벨이 다른 메뉴간 바로 이동이 가능하다.



Location history

- ▶ Location History 는 사용자의 현재 위치와 이전 경로를 함께 보여준다.
- ▶ 현재 위치를 기준으로 상위 메뉴까지 각각의 화면으로 바로 이동할 수 있는 링크 기능을 함께 제공하는 편이다.
- ▶ 화면 내에 고정적인 내비게이션 영역이 없거나 depth 가 깊은 메뉴 구조를 갖고 있을 때 적용된다.
- ▶ 주로 타이틀 영역 아래, 화면의 상단영역에 위치한다.



HOME > Menu01 > SubM01

기타 화면 이동

■ 홈

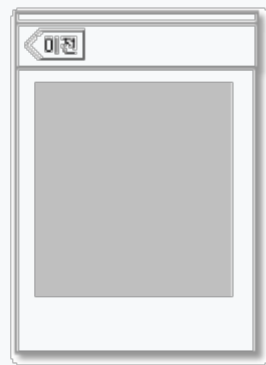
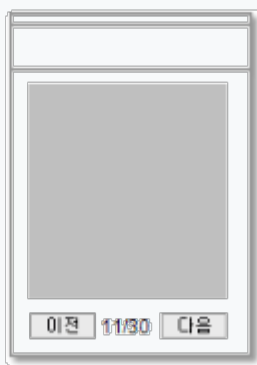
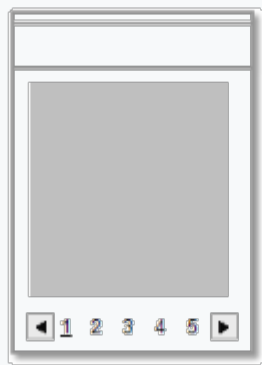
- ▶ 어느 화면이라도 홈 화면으로 이동할 수 있어야 한다.
- ▶ 홈 버튼 또는 로고영역 등 다양한 표현이 가능하다.

■ 페이지 내비게이션

- ▶ 페이지 단위 별로 이동할 경우 사용된다. 일반적으로 콘텐츠 영역 하단에 제공한다.

■ 이전(뒤로가기)/상위

- ▶ 이전화면으로의 이동은 브라우저나 디바이스에 기본적으로 제공할 수 있으나 상황에 따라 화면 내에 버튼형식으로 제공한다.

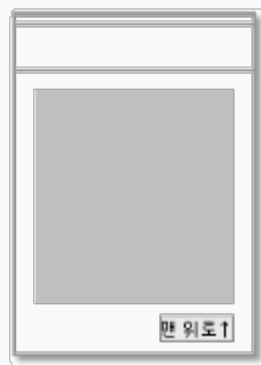
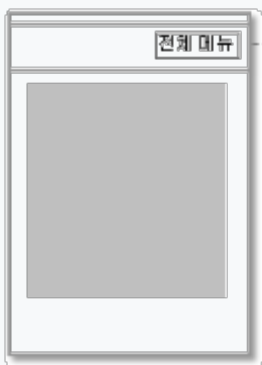


■ 전체 메뉴/서비스

- ▶ 최상위 단계의 메뉴로 바로 이동할 수 있는 기능을 버튼형식으로 제공하며 주로 상단에 배치한다.
- ▶ 홈 화면이 별도로 존재하지 않는 경우에 필수적으로 사용되어야 한다.

■ 맨 위 (TOP)

- ▶ 현재 화면 내에서 상위로 올라갈 때 쓰이며 주로 하단에 배치한다.
- ▶ 전체 화면을 한눈에 보기 어려울 정도로 콘텐츠가 많은 화면에 주로 사용된다.



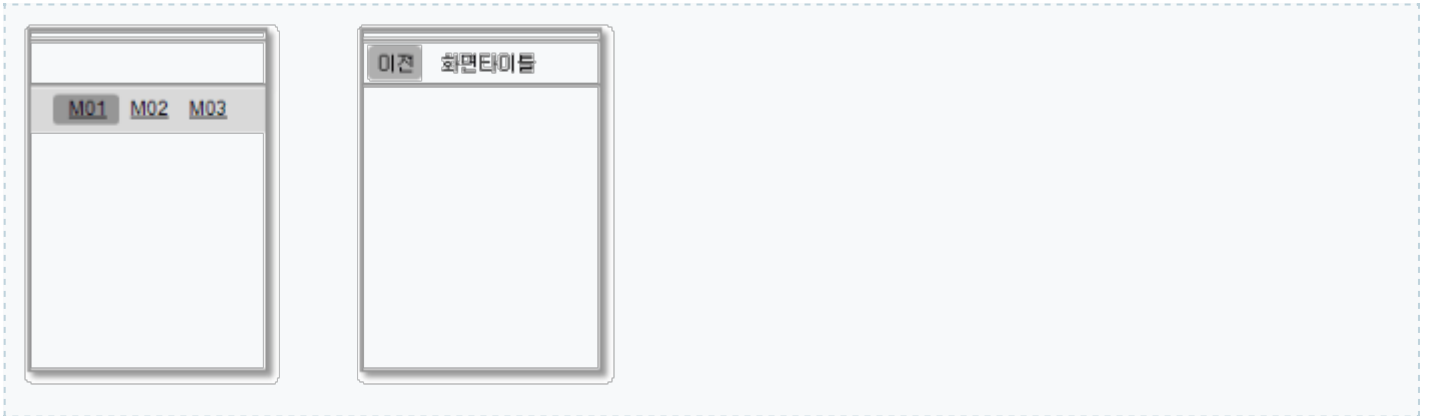
타이틀

정의

현재 화면에 대한 제목 영역이다.

적용가이드

- ▶ 항상 화면 상단에 나타나며 중앙정렬 혹은 경우에 따라 왼쪽 정렬이 가능하다.
- ▶ 화면 타이틀 영역은 비교적 좁은 영역에 표현되므로 글자 수의 제한이 필요하다.
- ▶ 탭 형식으로 표현된 메뉴의 경우 선택된 메뉴표현으로 타이틀을 대신할 수도 있다.
- ▶ 화면 타이틀 영역만을 위해 1줄을 할애하는 것은 권장하지 않는다.
- ▶ 가독성에 주의하여 표현한다. 장식적인 폰트나 현란한 색상을 사용하여 명확성을 해치지 않도록 한다.



메뉴

정의

메뉴는 사용자가 어느 화면에 있더라도 원하는 화면으로 쉽게 이동할 수 있도록 제공하는 콘텐츠 그룹이다.

적용가이드

- ▶ 각 메뉴 항목들은 하위항목이 서로 중복되지 않으며 명확하게 구분가능 해야 한다.
- ▶ 효율적이고 일관된 흐름을 위하여 메뉴 구조는 상하구조를 계층적으로 조직화한다.
- ▶ 메뉴는 고정된 위치에 일관성 있게 배치하며 한눈에 파악할 수 있도록 간단명료하게 제공한다.
- ▶ 메뉴는 본문 내용과 구분되어 터치 가능한 요소임을 시각적으로 구분할 수 있도록 스타일을 명확하게 정의한다.
- ▶ 메뉴를 조작할 때 메뉴의 선택한 상태나 이동 후 결과화면의 변화가 확실하게 반영되어야 한다.
- ▶ 화면에서 메뉴영역이 본문 영역보다 큰 비중을 차지하는 것은 권장하지 않는다. (메뉴의 레이아웃은 UI 패턴별가이드 참조)



탭바

정의

탭바는 복수개의 메뉴를 선택할 때 사용하는 컴포넌트이다.

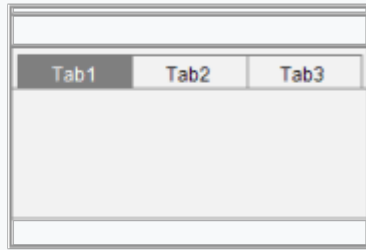
적용가이드

▶

- ▶ 1개의 메뉴로 이루어진 탭을 타이틀표현으로 사용하는 것은 바람직하지 않다.
- ▶ 탭은 최대 4개를 넘지 않는 것이 좋으며 5개 이상으로 표현할 때는 5번째 버튼에 more 버튼을 사용하여 접혀진 메뉴로 제공하거나, 리스트 메뉴의 형태로 표현하는 것이 좋다.
- ▶ 탭은 선택된 탭의 시각적 형태를 달리하여 선택된 탭을 확연히 구분할 수 있도록 해 주어야 한다.
- ▶ 태블릿의 경우 4개 이상의 표현이 가능하나 너무 많은 탭을 넣는 경우 복잡성을 증가시키므로 7개 이하를 권장한다.
- ▶ 탭바는 디바이스의 방향을 바꾸어도 개수나 스타일 표현이 달라지지 않아야 한다.
- ▶ 탭바의 타이틀에 아이콘을 사용하는 경우 아이콘과 텍스트를 함께 제공해 주어야 한다.



상단 탭표현(세로모드)



상단 탭표현(가로모드 변환시)

버튼

정의

버튼은 액션이나 내비게이션을 위해 사용하는 컴포넌트이다.

적용가이드

- ▶ 버튼은 터치 가능한 최소영역 40*40 Pixel 이상의 크기로 제작되어야 한다.
- ▶ 버튼을 탭 했을 때는 시각적으로 적절한 피드백을 제공해 주어야 한다.
- ▶ 중요한 버튼의 경우 크고 잘 보이는 위치에 표현한다.
- ▶ 버튼 명은 가능한 짧고, 직관적으로 의미전달이 가능해야 한다.
- ▶ 버튼 명은 한글을 기본으로 한다.
- ▶ 버튼의 레이블은 영문명을 제외하고는 붙여쓰기 하는 것이 좋다. (예: 신청서 작성→신청서작성)
- ▶ 버튼의 텍스트는 충분한 가독성을 확보할 수 있는 크기로 제공되어야 한다.
- ▶ 버튼의 형태는 클릭 가능한 상태로 보여야 한다.
- ▶ 버튼은 Default, Disabled, Focused, Pressed 의 4가지 상태를 가진다. (터치스크린 화면에서는 Focused 상태의 동작이 안 됨)
- ▶ 버튼의 정렬은 화면의 중앙이나 우측정렬이 가능하며 전체사이트에 페이지 별로 일관된 규칙성을 가지는 것이 좋다.
- ▶ 버튼의 배치는 중요한 순으로 좌측에서 우측으로 배치하는 것이 좋다.
- ▶ CSS3에서 추가된 항목 중 이전에는 이미지를 이용해야만 가능했던 Gradient, DropShadow, Round Corner 같은 효과들을 태그를 이용하여 만들 수 있으며 이 기능들을 이용하면 불필요한 이미지 사용을 줄일 수 있다.
- ▶ 액션의 결과가 부정적 영향을 미칠 수 있는 버튼은 색상을 달리하여 표현하고 추가로 확인하는 절차를 두는 것이 좋다. (예: 전체삭제 버튼 등)
- ▶ 현재 화면에서 선택이 가능한 버튼과 불가능한 버튼은 명확히 구분되도록 한다.



■ 적용가이드

- ▶ 제조사 디바이스별 기본적으로 제공되는 폰트가 다르기 때문에 가능하면 디바이스 별 기본 폰트를 사용하는 것을 권장한다.
- ▶ 같은 플랫폼에서 제공하는 폰트라도 버전에 따라 기본 자간과 볼드처리가능여부 등의 차이가 있으므로 가능하면 다양한 디바이스 환경에서 테스트 하는 것이 바람직하다.
- ▶ 모바일 기기의 특성상 사용자가 임의로 폰트의 크기나 서체를 바꿀 수 있다는 점을 감안하여야 한다. (iPhone은 설정 불가)
- ▶ 모바일 디바이스의 PPI는 PC모니터의 PPI보다 높기 때문에 같은 크기의 폰트를 적용하면 모바일에서는 더 작게 보인다는 점을 고려해야 한다. 따라서 최소 14px 이하의 폰트 사이즈는 지정하지 않을 것을 권장한다.
- ▶ 사용자가 반드시 읽어야 하는 내용의 콘텐츠는 본문텍스트 폰트확대 기능을 추가적으로 제공해 주는 것이 좋다.
- ▶ 폰트의 색상은 바탕색과 명도대비를 크게 하여 명확히 구분될 수 있도록 한다.
- ▶ 여러 줄의 콘텐츠는 충분한 줄 간격을 확보하여야 가독성을 해치지 않는다.
- ▶ 레이블링은 제한된 공간 안에 제공해야 하므로 최대한 간략하게 표현해야 한다.
- ▶ 레이블링은 한글표기를 기본으로 하며 고유명사의 경우 영문 그대로 표기한다. (News(x) 뉴스(o))
- ▶ 레이블링은 명사형이나 동사의 명사형, 수식어와 명사형으로 한다. 서술적인 표현이나 형용사는 레이블링으로 적합하지 않다.

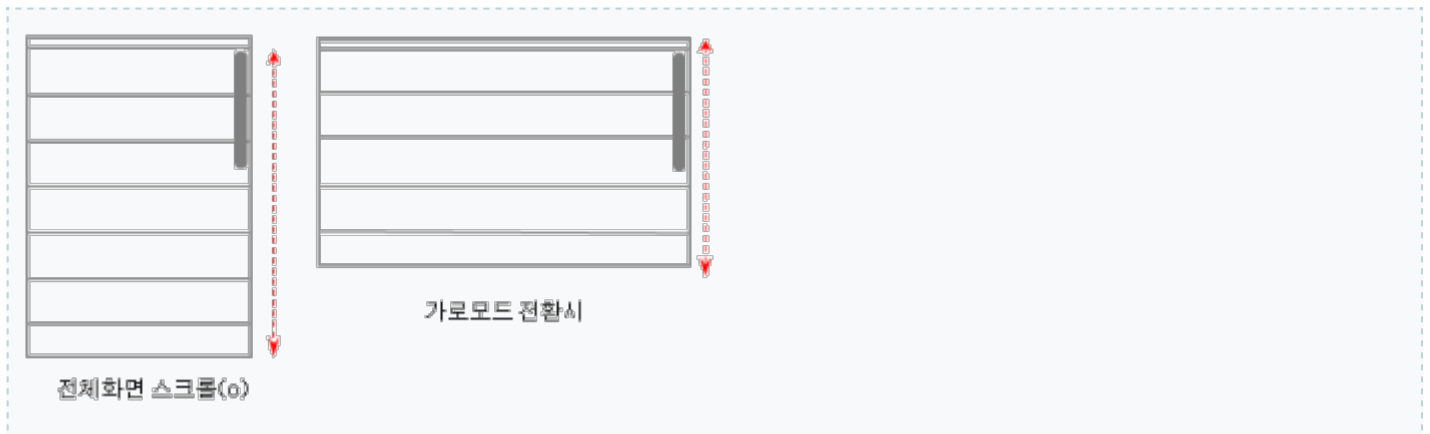
스크롤

■ 정의

스크롤은 콘텐츠의 내용이 현재 화면보다 더 길거나 넓은 경우 현재위치를 표시하기 위한 컨트롤이다.

■ 적용가이드

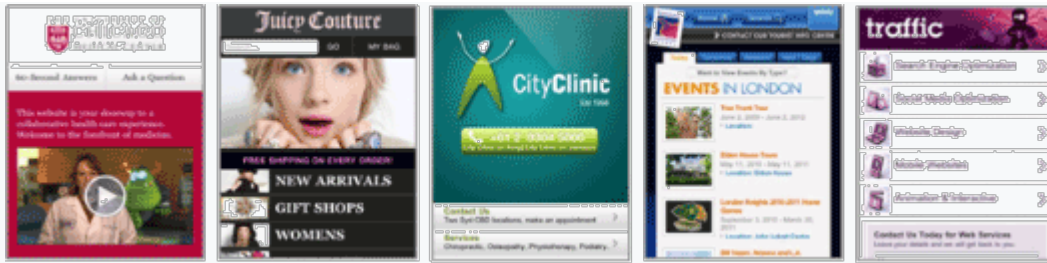
- ▶ 화면의 우측이나 하단에 바 형태로 나타난다.
- ▶ 기본스크롤이 나타나게 하는 것을 기본으로 한다.
- ▶ 스크롤은 세로스크롤이나 가로스크롤 중 한쪽 방향만 나타나게 하는 것이 바람직하다.
- ▶ 화면에 부분적으로 스크롤이 나타나게 하는 경우 전체 스크롤과 중복되어 나타나게 되면 동작이 어려우므로 권장하지 않는다.



컬러

■ 적용가이드

- ▶ 일반적으로 컬러는 아이덴티티의 중요한 역할을 수행하게 된다.
- ▶ 아이덴티티의 일관성 유지를 위하여 모든 페이지에 발생하는 이미지 요소는 컬러의 구성 및 적용범위를 규정하여 적용하거나 응용하여 사용하도록 한다.
- ▶ 컬러시스템 적용이 어려운 사진 이미지는 컬러 시스템의 적용 범위에서 제외하도록 한다.
- ▶ 필요 시 특별한 주제의 강조나 특별한 콘텐츠 내용의 표현을 위하여 특수한 컬러를 도입하여 사용할 수 있도록 한다.
- ▶ 시각적인 피로도가 높은 컬러의 사용은 자제하도록 한다.
- ▶ 텍스트 색상은 배경색과 충분한 대비를 두어 글자가 눈에 잘 들어오도록 한다.
- ▶ 본문 내용이 들어가는 영역의 배경에는 패턴이나 텍스처 적용을 자제한다.
- ▶ 지나치게 다양한 색상을 적용하면 시선이 분산되어 혼란스러우므로 색상 가지 수를 제한한다.



페이지 표현

정의

페이지 Indicator는 얼마나 많은 화면이 있고 현재 보여지는 것은 어떤 페이지인지 보여준다.

적용가이드

- ▶ 페이지순서는 왼쪽에서 오른쪽 순으로 표현된다.
- ▶ 현재 선택된 페이지에 대해서는 시각적인 차별화를 통해 명확히 구분되도록 해야 한다.
- ▶ 사용자는 좌우 버튼이나 페이지를 직접 탭하여 내비게이션 할 수 있다.
- ▶ 화면이 동등한 레벨일 때 표현할 수 있는 방법이며, 계층구조를 보여주기에는 적합하지 않다.
- ▶ Dot 으로 표현된 페이지 Indicator 는 제한된 수의 페이지만을 표현할 수 있다.
- ▶ 더 많은 페이지를 표현해야 하는 경우는 숫자로 표현된 페이지 내비게이션을 활용하는 것을 권장한다.
- ▶ 페이지 Indicator 처럼 사용자가 작은 영역을 탭하는 경우는 좌우 여백을 충분히 확보하여야 오작동을 줄일 수 있다.



숫자형 Page Indicator



Dot형 Page Indicator

팝업

정의

팝업은 현재의 창에서 처리할 수 없는 독립된 내용을 추가적인 창을 띄워서 보여주기 위해 사용된다.

적용가이드

- ▶ 모바일에서 불필요한 팝업은 최대한 사용을 자제하는 것이 좋다.
- ▶ 선택이나 확인 창을 과도하게 사용하지 않아야 한다.

체크박스

정의

체크박스는 여러 개의 메뉴 중 복수선택이 가능할 때 사용되는 컨트롤이다.

적용가이드

- ▶ Default 상태에서는 아무것도 선택되지 않은 상태로 나타나야 한다.
- ▶ 체크박스를 탭하면 V마크로 표현되며, 기선택된 체크박스를 터치하면 사라진다.
- ▶ 전체선택, 전체해제가 필요한 경우는 상단영역에 해당버튼을 두어 컨트롤 하도록 한다.
- ▶ 지나치게 많은 선택항목을 제시하여 스크롤영역 하단으로 내려가는 것은 사용성을 저하시키므로 항목을 최소화 하는 것을 권장한다.

라디오버튼

정의

라디오버튼은 여러 개의 메뉴 중에서 하나의 메뉴만 선택 가능할 때 사용되는 컨트롤이다.

적용가이드

- ▶ Default 상태에서는 현재 설정된 값이 선택된 상태로 나타난다.

- ▶ 라디오 버튼은 선택항목을 한 화면에 표시해야 한다.
- ▶ 선택되지 않은 라디오버튼을 탭하면 해당 버튼이 체크되고 기 선택된 버튼은 자동으로 해제된다.
- ▶ 라디오 버튼 조작 후 저장버튼을 두지 않아도 자동으로 저장되도록 한다.

아이콘 & 이미지

■ 적용가이드

- ▶ 아이콘은 텍스트 레이블을 사용하기에 공간이 좁거나 아이콘으로 의미전달에 더 명확할 때 사용한다.
- ▶ 이미지사용은 가능한 최소한의 요소만을 적용하여 경량화된 디자인을 제공하는 것이 좋다.
- ▶ 이미지 포맷은 PNG, JPG, GIF 포맷을 권장한다.
- ▶ 아이콘과 이미지는 서비스의 Look&Feel을 형성하는 요소이므로 각 서비스의 Identity를 살려 제작한다.
- ▶ 아이콘 제작 시 사용자가 내용을 직관적으로 식별할 수 있는 메타포를 사용해야 한다.
- ▶ 아이콘은 전체 디자인과 조화를 이룰 수 있도록 디자인 되어야 한다.
- ▶ 여러 개의 아이콘이 사용되는 경우, 아이콘들 간의 스타일과 크기의 일관성을 가지도록 제작한다.
- ▶ 간결하게 표현하며 작은 이미지에 과도한 장식이나 여러 가지 컬러를 적용하지 않는 것이 좋다.
- ▶ 타인의 저작권을 침해하는 사진 이미지나 아이콘은 사용하지 않는다.
- ▶ 공공서비스의 다양한 사용자에 이해를 높이기 위해서 아이콘은 가능하면 텍스트 레이블과 함께 사용하는 것을 권장한다.

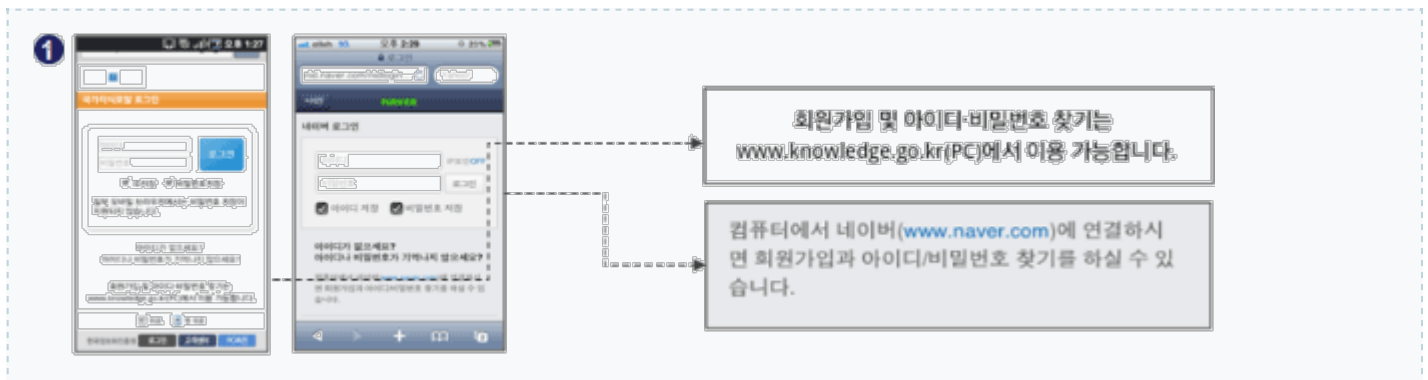
입력 폼

■ 정의

사용자가 화면내 키보드를 이용하여 정보를 넣는 영역이다.

■ 적용가이드

- ▶ 모바일은 일반 PC보다 상대적으로 입력방식이 어려우므로, 사용자가 입력하는 화면은 되도록 자제하며, 불가피하게 입력화면이 필요할 경우 항목을 단순화하고 한 화면에 제공하는 입력 항목이 길어지지 않게 여러 단계를 나누어서 입력하도록 한다. 입력화면에서는 스크롤 사용을 자제한다.
- ▶ 직접 입력보다는 사용자에게 옵션을 보여주어 선택하는 방식을 제공하는 것이 바람직하다. 빈도수가 높은 옵션이나 최근에 사용한 옵션이 디폴트 상태가 되도록 한다.
- ▶ 필요한 입력 콘텐츠가 일반 웹에 존재할 경우 PC버전을 사용하도록 유도한다. ①



동영상/멀티미디어

■ 적용가이드

- ▶ GIF Animation과 Flash의 경우 Platform별로 지원하지 않는 경우가 많으므로 사용을 지양 한다. Apple계열의 제품들은 Flash를 지원하지 않고 있으며 안드로이드도 상위 버전에서만 지원이 가능하다.
- ▶ 동영상 정보는 대체 텍스트 또는 설명 글을 같이 제공한다.

OS (Platform)	브라우저	Flash	GIF Animation	Multimedia
Android (v2.2)	Chrome	O	O	MP4, AVI, WMV, ASF (디바이스별로 다름)
iOS	Safari	X	X	MP4

OS (Platform)	브라우저	Flash	GIF Animation	Multimedia
Android (v2.2)	Chrome	O	O	MP4, AVI, WMV, ASF (디바이스별로 다름)

Progress bar

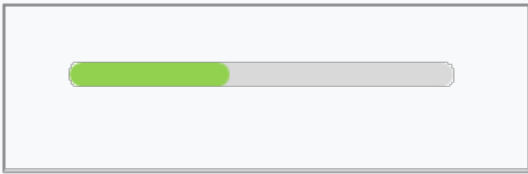
정의

프로그레스 바는 작업의 진행을 나타내는 바 형태의 컨트롤이다.

데이터의 다운로드나 전송시에 완료시간을 예측할 수 있는 상황에 사용한다.

적용가이드

- ▶ 프로그레스 바는 좌에서 우측 방향으로 진행한다.
- ▶ 여러 개의 작업이 순차적으로 진행되는 경우 프로그레스 바는 각 작업당 1회씩 나타난다.
- ▶ 작업시간이 2초 미만인 경우는 프로그레스 바를 생략할 수 있다.
- ▶ 진행 중 취소버튼을 함께 제공할 수 있다.



Progress bar

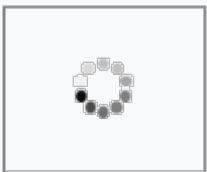
Progress indicator

정의

프로그레스 인디케이터는 콘텐츠가 로딩될 때 완료시점을 예측할 수 없는 상황에 진행 중인 상태를 보여주는 컨트롤이다.

적용가이드

- ▶ 콘텐츠 영역에서 이미지 등 로딩시간이 오래 걸리는 이미지가 있는 경우 텍스트를 먼저 표시하고 Progress indicator를 사용하는 것이 좋다.
- ▶ 프로그레스 인디케이터는 사용자와 상호작용하지 않는다.



Progress Indicator

드롭다운/피커

정의

두 가지 모두 선택 가능한 여러 항목 중 하나를 선택할 때 사용하는 컨트롤이다.

적용가이드

- ▶ 옵션바나 콘텐츠 영역을 탭 하면 하단에 나타난다.
 - ▶ 현재 설정 값과 선택할 수 있는 다른 항목을 보면서 선택할 수 있다.
- 탭 조작을 통해 콘텐츠를 선택하면 드롭다운이나 피커가 닫히지 않은 상태에서 선택항목에 선택된 내용이 표기된다.
- 드롭 박스 선택항목이 많은 경우 박스 안에 스크롤이 생겨 하단으로 메뉴가 나타날 수 있다.



화면해상도의 정의

미디어 쿼리 정의

모바일 기기별로 스타일을 다르게 적용해 주어야 한다.

- 모바일 기기별 스타일을 제공한다.

```
<link media="only screen and (max-device-width: 480px)" href="small-device.css" type="text/css" rel="stylesheet">
<link media="screen and (min-device-width: 481px)" href="not-small-device.css" type="text/css" rel="stylesheet">
```

- 가로, 세로 모드로 구분할 경우 각각의 다른 스타일로 제공한다.

```
<link rel="stylesheet" media="all and (orientation:portrait)" href="portrait.css">
<link rel="stylesheet" media="all and (orientation:landscape)" href="landscape.css">
```

viewport 메타 태그의 정의

모바일 브라우저가 각각 다르기 때문에 이들 브라우저의 크기 별로 적절하게 표현하기 위해서 사용하는 방법으로써 viewport 메타 태그를 head 영역에 삽입하여 문제점을 해결할 수 있습니다.

- viewport의 가로 크기를 디바이스 가로 크기로 설정할 경우

```
<meta name = "viewport" content = "width = device-width" />
```

- viewport의 기본 확대비율을 100%로 설정할 경우

```
<meta name = "viewport" content = "initial-scale = 1.0" />
```

- viewport의 기본 확대비율을 230% 및 사용자 확대 기능 불가로 설정할 경우

```
<meta name = "viewport" content = "initial-scale = 2.3, user-scalable = no" />
```

- 아이폰을 기준으로 옴니아 viewport 설정할 경우

```
<meta name="viewport" content="width=device-width; initial-scale=0.75; maximum-scale=0.75;" />
```

- 사파리, 오페라, 크롬 브라우저를 모두 충족하는 경우

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

■ viewport가 지원되지 않을 경우

viewport는 모든 모바일 브라우저에서 지원되지 않습니다. 이러한 경우에는 다른 meta tag를 사용하도록 합니다.

```
<meta name="HandheldFriendly" content="true" />  
<meta name="MobileOptimized" content="320" />
```

HTML5와 CSS3.0 기본 활용

전자정부 모바일 표준프레임워크는 HTML5 구조와 문법, CSS3.0 기본 활용에 관한 디자인 가이드를 제공한다.

사용자 경험 (UX) 선언

전자정부 모바일은 HTML5 의 DOCTYPE 으로 선언하여야 한다.

문서에 사용된 문자 인코딩을 간소화된 구문을 통해 meta 태그를 사용할 수 있다.문서의 인코딩은 “UTF-8” 사용을 권장한다.

```
<!DOCTYPE html>
<html>
  <head>
    <title>eGovFrame</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no"/>
    <link rel="stylesheet" href="/css/egovframework/mbil/cmm/jquery.mobile-1.0b3.min.css"/>
    <link rel="stylesheet" href="/css/egovframework/mbil/cmm/EgovMobile.css" />
    <script src="/js/egovframework/mbil/cmm/jquery-1.6.2.min.js"></script>
    <script src="/js/egovframework/mbil/cmm/jquery.mobile-1.0b3.min.js"></script>
    <script src="/js/egovframework/mbil/cmm/EgovMobile.js"></script>
  </head>
  <body>
    ...
  </body>
</html>
```

모바일 페이지 구조

모바일 의 Page 구조는 DIV 태그의 data-role="page" 속성을 이용하여 표출 한다.

Page 내부의 구조는 마찬가지로 data-role 의 header, contents, footer 의 속성을 이용하여 표출 한다.

```
<body>
  <!-- 모바일 페이지 start -->
  <div data-role="page">

    <!-- header start -->
    <div data-role="header">
      <h1>header Title</h1>
    </div>
    <!-- header end -->

    <!-- content start -->
    <div data-role="content">
      <p>전자정부 모바일 표준프레임워크 default Page 화면 입니다.</p>
    </div>
    <!-- content end -->

    <!-- footer start -->
    <div data-role="footer">
      <h4>footer title</h4>
    </div>
    <!-- footer end -->

  </div>
```



```
<!-- 모바일 페이지 end -->
</body>
```

CSS3.0 기본 활용

개별적으로 스타일을 지정하는 것보다는 클래스(class)와 외부 CSS 파일을 이용하는 것이 문서 전체의 일관성은 물론 스타일 변경시에도 작업의 효율성을 높여줍니다.

CSS 문법과 클래스(class) 와 아이디 (id)

CSS 구문을 이해하기 위해서는 선택자(selector), 속성(property) 그리고 속성값(value) 이 3가지를 이해하여야 합니다.

```
선택자 { 속성: 값(단위); } //주석
```

클래스를 독립적으로 지정하는 방법이 좋으며 여러 클래스를 사용가능하다.

```
<h3 class="red">소제목에도 red 클래스를 지정합니다.</h3>
```

아이디는 고유성을 가지므로 한 문서에 한번만 사용할 수 있다.

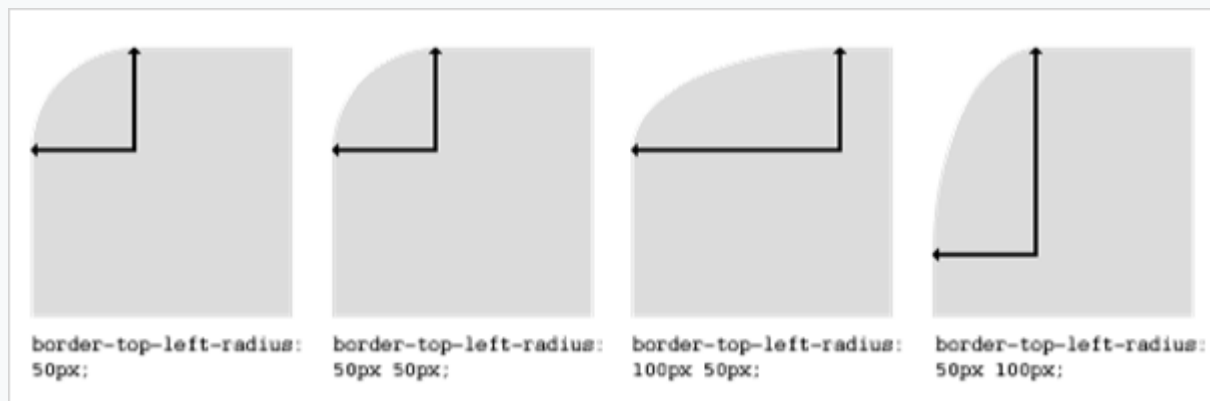
```
<p id="red">이 문단의 id 는 red 입니다.</p>
```

CSS3.0 기본 기능

■ 둥근 모서리

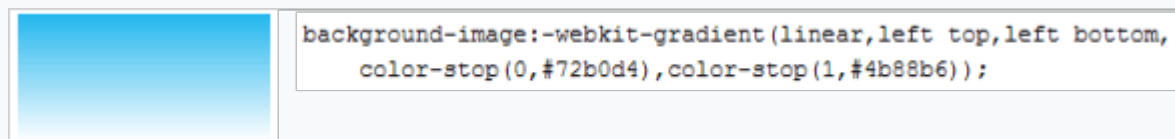
이미지를 사용하지 않고, 둥근 모서리 또는 테두리와 버튼을 생성할 수 있다.

둥근 사각형 (Rounded Rectangle): border-radius 속성 값의 비율로 모서리에 곡선 적용



■ 그라데이션(Gradients)

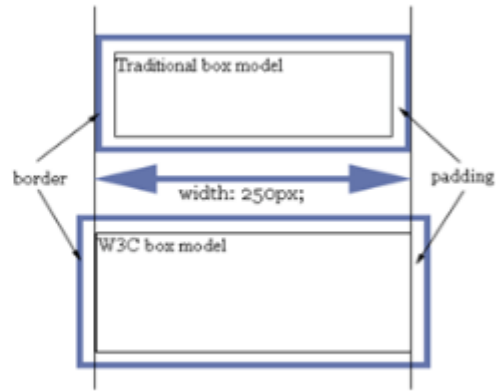
- ▶ 배경색상을 변경할 때 사용할 수 있다.
linear (선형 gradient) 와 radial(원형 gradient) 두 가지로 나눌 수 있다.
- ▶ CSS 에서 Gradient 효과가 가능하도록 Webkit 에서 지원한다.



■ box-sizing을 내가 원하는 대로 표현할 수 있다.

min/max-width/height와 함께 사용 할 수 없다.

```
border/content-box
.box {
box-sizing: content-box;
-moz-box-sizing: content-box;
-ms-box-sizing: content-box;
-webkit-box-sizing: content-box;
}
```



■ 다양한 투명도를 사용할 수 있다.

```
#trans {
opacity: 0.4;
-moz-opacity: 0.4;
-webkit-opacity: 0.4;
filter:alpha(opacity=40);
}
```

Regular image:



The same image with transparency:



■ 다운로드하여 웹 폰트를 가져올 수 있다.

웹 폰트의 경로명을 통해 폰트를 쉽게 추가할 수 있다.

```
@font-face {font-family: "yunache"; src: url(http://.../yunache.ttf) format("truetype");}
h1 { font-family: "yunache", sans-serif; }
```

■ CSS3에서 텍스트 그림자 속성은 텍스트에 그림자를 적용할 수 있다.

수평 그림자, 수직 그림자, 흐림 효과와 그림자의 색상을 지정할 수 있다.

사용법은 아래와 같다.

Text shadow effect!

```
h1{text-shadow: 5px 5px 5px #FF0000;}
```

■ 상자에 그림자를 추가하여 사용할 수 있다.

- ▶ 대부분의 요소의(element) 프레임에 그림자 효과를 적용한다.
- ▶ box-shadow: offsetX offsetY blurRadius color

offsetX : X축으로 +/- 픽셀 위치 : 박스(텍스트) 왼쪽(-) 또는 오른쪽(+)

offsetY : Y축으로 +/- 픽셀 위치 : 박스(텍스트) 위쪽(-) 또는 아래쪽(+)

blurRadius : 흐려짐 정도

color : 색상

사용법은 아래와 같다.



```
div {  
-moz-box-shadow: 10px 10px 5px #888888; /* Firefox */  
-webkit-box-shadow: 10px 10px 5px #888888; /* Safari and Chrome */  
box-shadow: 10px 10px 5px #888888;}
```

■ Word-Wrap 속성을 이용하여 강제로 줄 바꾸기

숫자나 영문에 띄어쓰기 없이 쓰여진 문자는 요소의 너비를 무시하고 다음 공백이 나타날 때 까지 줄바꿈 되지 않을 때 사용할 수 있다.



사용법은 아래와 같다.

```
.break-word {word-wrap: break-word;}
```

■ 벤더(vendor) 별 확장 속성 및 브라우저 엔진

- ▶ 대부분의 브라우저 벤더들은 자신들의 브라우저가 현재 지원하는 속성이 표준과 상이하거나 변경 될 수 있다라고 생각하고 벤더 확장 속성을 만들었다. 실제로 속성을 지원하지만 100% 표준스펙이 나온 상태가 아니기 때문에 개선점이나 버그 발생시 피드백을 쉽게 하기 위해 만든 것이다.
- ▶ 브라우저 /벤더확장 속성/브라우저 엔진
사용에는 아래와 같다.

사파리 S5/ -webkit- / webkit



크롬 C8/ -webkit- / webkit

파이어폭스 F3.6/ -moz- / Gecko

오페라 O11 / -o- / presto

인터넷익스플러우 IE9 / -ms-, filter / Tasman

지원 플랫폼

컴포넌트				
지원버전	4.3 ↑	2.3 ↑	5.0 ↑	11.50 ↑
Button	O	O	O	O
Panel	O	O	O	O
Internal/External Link	O	O	O	O
Label/Text	O	O	O	O
Tabs	O	O	O	O
Form	O	O	O	O
Menu	O	O	O	O
Processing Dialog / Bar	O	O	O	O
Dialog	O	O	O	O
Grid View	O	O	O	O
Table / List View	O	O	O	O
Date / Time Picker	O	O	O	O
Check/ Radio	O	O	O	O
Icon	O	O	O	O
Selector / Switch	O	O	O	O
Collapsible Block	O	O	O	O

▶ 기준일 : 2011.11.15

▶ 행정안전부 고시 "제 2010-40호" 「전자정부서비스 호환성 준수지침」의 모바일 웹 방식 서비스 개발을 위한 기술 표준 지침에 따라 최소 3종 이상의 웹 브라우저에서 동등한 서비스 제공 기능을 지원 합니다.

제약사항

- Opera Mobile 11.50 이하 버전의 브라우저의 경우 UX Component의 gradient 효과가 적용되지 않아 일부 컴포넌트의 모양이 다를 수 있습니다.
- Internet Explorer 8 이하 버전의 브라우저의 경우 정상지원되지 않습니다.

지원 사양

컴포넌트	제공기능	
Button	설명	명령 수행, 옵션 선택, 다른 대화 상자 열기 등에 사용하는 컴포넌트 제공
	형태	둥근 형(radius: 1em), 사각형(radius: 0em)
	배치	vertical group, horizontal group
	색상	검정, 파랑, 회색, 흰색, 노랑, 빨강, 초록
	높이	normal(39px), small (28px)
	넓이	화면에 맞게, 텍스트에 맞게
Panel	설명	Header/footer 와 함께 페이지를 구성하는 요소
	무늬	격자 형태 무늬 제공
	색상	검정, 회색, 연한회색, 흰색, 노랑, 빨강, 초록
Internal / External Link	설명	표준 링크 기능을 제공 하며 기본적으로 Ajax 를 사용한 링크 방식 제공
	링크	페이지 내부링크,도메인 내부 링크,외부 링크,이메일 링크,폰 링크,에러 페이지 링크
Label / Text	설명	색상, 배치, 크기,폰트 를 지정 할 수 있는 가이드 제공
	색상	초록, 빨강, 파랑
	배치	왼쪽, 중간, 오른쪽
	크기	15px, 25px, 30px
	폰트	helvetica, verdana, tahoma
Tabs	설명	Header와 footer 에 사용되며 탭 버튼으로 문서간 이동 기능 제공
	형태	round tab(radius: 0.250em), normal tab(radius: 0em)
	배치	1, 1/2, 1/3, 1/4, 1/5, 1/2 다중행 tab
	색상	검정, 파랑, 회색, 흰색, 노랑, 빨강, 초록
Form	설명	HTML Form 요소를 모바일 환경에 최적화하여 제공
	요소	Text inputs, Search inputs, Sliders, Switches, Radio buttons, Check boxes, Selectors
	색상	검정, 파랑, 회색, 흰색, 노랑, 빨강, 초록
Menu	설명	Dialog, Grid, List, Collapsible 컴포넌트를 사용하여 메뉴 구성 기능 제공
	효과	slide, slideup, slidedown, pop, fade, flip
	형태	Dialog, Grid, List, Collapsible
	색상	검정, 파랑, 회색, 흰색, 노랑, 빨강, 초록
Processing Dialog / Bar	설명	페이지 전환간 진행 상태를 확인 할 수 있는 Progress Dialog/Bar 제공
	형태	Processing Dialog, Processing Bar
	색상	검정, 파랑, 회색, 흰색, 노랑, 빨강, 초록
Dialog	설명	사용자와 상호 작용 할 수 있는 Dialog 기능을 제공
	형태	Dialog, Action Sheet, Overlay, Alert, Prompt, Confirm
	색상	검정, 파랑, 회색, 흰색, 노랑, 빨강, 초록
Grid View	설명	Grid 형태로 콘텐츠를 배치 할 수 있는 컴포넌트 제공
	배치	1/2, 1/3, 1/4, 1/5, 가변 Grid View
Table / List View	설명	Table/List 형태로 콘텐츠를 배치 할 수 있는 컴포넌트 제공
	형태	Read-only list, Link list
	기능	Nested List, Numbered List, Split Button, List Divider, Count Bubble, Thumbnail, List icon, Content Formatting, Search Filter Bar
	색상	검정, 파랑, 회색, 흰색, 노랑, 빨강, 초록
Date / Time Picker	설명	날짜와 시간을 선택 할 수 있는 Picker 제공

	형태	Android Date Picker, Popup Calendar, Android Time Picker
Check/ Radio	설명	Check/Radio 형태로 항목을 선택할 수 있는 기능을 제공
	배치	vertical group, horizontal group
	색상	검정, 파랑, 회색, 흰색, 노랑, 빨강, 초록
Icon	설명	모바일 어플리케이션에 가장 많이 사용되는 아이콘을 제공
	형태	arrow-l, arrow-r, arrow-u, arrow-d, delete, plus, minus, check, gear, refresh, forward, back, grid, star, alert, info, search, home, phone, mail
	색상	검정, 파랑, 회색, 흰색, 노랑, 빨강, 초록
Selector / Switch	설명	Selector/Switch 형태로 항목을 선택 할 수 있는 기능을 제공
	효과	pop-up, list
	기능	다중선택, 단일 선택
	모양	둥근 형(radius: 1em) 사각형(radius: 0em)
	넓이	화면에 맞게 , 텍스트에 맞게
	효과	Shadow 적용, Shadow 제거
	형태	비그룹, 그룹
	색상	검정, 파랑, 회색, 흰색, 노랑, 빨강, 초록
Collapsible Block	설명	콘텐츠 영역을 접었다 펼 수 있는 컨트롤 기능 제공
	형태	normal, Group, Nested
	색상	검정, 파랑, 회색, 흰색, 노랑, 빨강, 초록

▶ 사용자 제공 UX의 기능은 위 표에 정의된 기능을 제공하며, 이외의 확장은 제한 없음

Configuring Defaults

전자정부 모바일 표준프레임워크는 모바일 어플리케이션이 시작 될때 각종 초기화 작업을 수행 할 수 있게 mobileinit 이벤트를 통하여 기본 환경 설정을 정의 할수 있도록 한다.

The Mobileinit Event

mobileinit 이벤트는 Page가 시작되자마자 발생하는 이벤트로서 첫 초기화 작업 시 실행되며 함수 내부에 여러 이벤트를 적용하여 사용 할 수 있다.

```
$(document).bind("mobileinit", function(){
    //apply overrides here
});
```

이 이벤트는 실행 즉시 발생 하므로 jquerymobile.js가 로드 되기 전에 바이딩 되어야 한다. 다시 말해 mobileinit 이벤트의 위치는 jquery 라이브러리와 jquerymobile 라이브러리 사이에 위치 해야한다.

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

기본 환경설정을 변경하는 방법에는 \$.extend 메소드를 이용하여 여러 속성을 한 번에 변경하거나

```
$(document).bind("mobileinit", function(){ $.extend(
    $.mobile , {
        foo: bar
    });
});
```

.mobile 객체를 사용하여 개별적으로 변경할 수도 있다.

```
$(document).bind("mobileinit", function(){
    $.mobile.foo = bar;
});
```

Configurable Options

다음의 기본 환경설정들은 \$.mobile 객체를 통해 재설정이 가능하다 :

ns (string, default: ""):

namespace는 data-role과 같은 'data-*' 형식의 속성에 사용된다. namespace는 default인 빈 문자열(blank string) 부터 사용자 지정의 어떤 값으로도 지정이 가능하다. namespace를 명확하게 사용하려면 중간에 '-'(trailing dash)를 추가하는 것이 좋다.

NOTE: 'data-*' 형식의 namespace를 사용할 때는 테마 CSS 파일의 선택자(selector)를 아래와 같이 개별적으로 Update/Override 해야 한다

```
.ui-mobile [data- mynamespace- role=page], .ui-mobile [data- mynamespace- role=dialog], .ui-page { ...
```

autoInitializePage (boolean, default: true):

DOM이 실행될 때 \$.mobile.initializePage를 호출하여 자동 초기화를 진행하는데 autoInitializePage를 false로 선언 할 경우 는 자

동 페이지 초기화를 하지 않는다.

subPageUrlKey (string, default: "ui-page"):

URL parameter는 Nested List 처럼 위젯에 의해서 생성되는 하위 페이지(subpage) 참조를 위해 사용되며 'example.html&ui-page=subpageIdentifier' 형식으로 변경된다. '&ui-page=' 앞의 hash segment 는 프레임워크에서 하위 페이지가 존재하는 URL 에 대한 Ajax 요청을 만들기 위해 사용된다.

activePageClass (string, default: "ui-page-active"):

현재 보여지고 있거나 트랜지션(transition) 중인 페이지에 할당되는 클래스이다.

activeBtnClass (string, default: "ui-page-active"):

현재 활성화 되어있는 버튼에 할당되는 클래스이다.

ajaxEnabled (boolean, default: true):

모든 링크 이동이나 폼 전송은 기본적으로 Ajax를 기반으로 하고 있다 Ajax가 아니라 일반 방식으로 페이지 이동을 처리하고 싶다면 이 값을 false로 지정한다.

hashListeningEnabled (boolean, default: true):

history의 변경을 자동적으로 처리 한다. hashListeningEnabled 옵션을 'disabled' 로 설정하면 history의 변경을 자동으로 처리하지 않게 된다.

defaultPageTransition (string, default: 'slide'):

Ajax 방식을 사용하는 페이지 전환에 관여하는 기본 환경설정을 변경한다. defaultPageTransition 옵션을 'none'으로 설정하면 화면전환 효과가 적용되지 않는다.

touchOverflowEnabled (boolean, default: false):

디바이스의 스크롤 및 fixed toolbar 기능을 사용하기 위하여 overflow-scrolling: touch 지원하는 여부를 설정한다.

defaultDialogTransition (string, default: 'pop'):

다이얼로그에서 Ajax 방식을 통한 페이지 전환에 관여하는 기본 환경설정을 변경한다. defaultDialogTransition 옵션을 'none'으로 설정하면 화면전환 효과가 적용되지 않는다.

minScrollBack (string, default: 150):

스크롤을 시작하는 최소거리를 설정한다.

loadingMessage (string, default: "loading"):

로딩시 보여지는 로딩 메시지를 변경한다. 'false'로 설정하면 로딩 메시지가 나타나지 않는다.

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

Ajax 방식의 페이지이동에서 페이지 로드를 실패 했을 경우 나타나는 메시지를 변경한다.

I Events

전자정부 모바일 표준프레임워크는 스마트기반 모바일 환경에 적합한 이벤트를 제공한다. Touch, Mouse, Window 영역의 다양한 이벤트를 지원가능 여부에 따라 선택적으로 이용하기 때문에 모바일 환경과 데스크톱(Desktop) 환경 모두에서 사용 가능하다. live() 또는 bind() 메소드를 이용하여 여러 이벤트를 함께 사용 할 수 있다.

아래 이벤트는 pageCreate()에 live() 또는 bind() 해야 사용 할 수 있다.

Touch Events

tap

Touch 이벤트가 끝나는 즉시 발생하는 이벤트이다.

taphold

tap을 일정 시간 이상 지속했을 때 발생하는 이벤트이다.

swipe

1 초 이내에 30px이상의 수평 방향이나 20px 이상의 수직 방향으로 드래그(drag)되면 발생하는 이벤트이다.

scrollSupressionThreshold (default: 10px) – default 값 이상 으로 수평방향으로 이벤트가 발생하면 스크롤이 발생되지 않도록 한다.

durationThreshold (default: 1000ms) – default 값 이상 swipe이 지속되면 기능이 작동하지 않는다.

horizontalDistanceThreshold (default: 30px) – default 값 이상 수평방향으로 이벤트가 발생해야한다.

verticalDistanceThreshold (default: 75px) – default 값 이하로 수직방향으로 이벤트가 발생해야한다

swipeleft

swipe 이벤트가 왼쪽으로 일어났을 때 발생하는 이벤트이다.

swiperight

swipe 이벤트가 오른쪽으로 일어났을 때 발생하는 이벤트이다.

Orientation change event

orientationchange

기기의 방향이 수평 또는 수직으로 바뀌었을 때 발생하는 이벤트이다. orientationChange 이벤트가 지원되지 않을 경우에는 resize 이벤트가 자동으로 bind 된다.

Scroll events

scrollstart

스크롤(scroll)이 시작되면 발생하는 이벤트이다. (iOS 기기는 스크롤 중에는 DOM 을 변경하지 않고 queue 에 저장해두었다가 스크롤이 끝난 후에 변경한다.)

scrollstop

스크롤이 끝나면 발생하는 이벤트이다.

Page Events

페이지가 나타나기 이전(pagebeforeshow)과 나타난 순간(pageshow), 가려지기 이전(pagebeforehide)과 가려진 순간(pagehide) 페이지 이벤트가 발생한다.

pagebeforeshow

화면전환이 일어나기 전, 또는 페이지가 보여지기 전에 매번 발생하는 이벤트이다.

pagebeforehide

화면전환이 일어나기 전, 또는 페이지가 숨겨지기 전에 매번 발생하는 이벤트이다.

pageshow

화면전환이 완료되었거나 페이지가 보여진 후에 매번 발생하는 이벤트이다..

pagehide

화면전환이 완료되었거나 페이지가 가려진 후에 매번 발생하는 이벤트이다.

페이지 이벤트는 이전 페이지와 다음 페이지의 존재 유무와 이벤트 종류에 따라 이전 페이지(prevPage)와 다음페이지(nextPage)에 대한 정보를 갖는다. (첫 번째 페이지는 이전 페이지에 대한 정보가 없기 때문에 null 객체를 참조한다.)

```
$('#div').live('pageshow', function(event, ui) {  
    alert('This page was just hidden: '+ ui.prevPage);  
});  
  
$('#div').live('pagehide', function(event, ui) {  
    alert('This page was just shown: '+ ui.nextPage);  
});
```

최초 페이지가 로드될 때 페이지 이벤트에 관련된 핸들러를 사용하기 위해서는 참조된 Javascript 가 실행되기 전에 bind 시켜야 한다. Global Configuration 에서 설명한 것처럼 mobileinit 핸들러를 이용하면 이와 같은 설정을 할 수 있다.

Virtual mouse events

마우스와 터치 이벤트를 둘다 인식 할 수 있는 virtual click을 제공한다. 이는 기존 웹브라우저 환경에서 발생하는 기본 마우스 이벤트(mouseup, mousedown, mousemove, click)가 모바일 환경에서도 지원이 되며 동일 자원에 다중 이벤트가 발생할 시 각각 다른 이벤트로 인식한다.

vmouseover

터치 이벤트 또는 mouseover가 발생할 때 나타나는 이벤트이다.

vmousedown

터치 이벤트 또는 mousedown이 발생할 때 나타나는 이벤트이다.

`vmousemove`

터치 이벤트 또는 `mousemove`가 발생할 때 나타나는 이벤트이다.

`vmouseup`

터치 이벤트 또는 `mouseup`이 발생할 때 나타나는 이벤트이다.

`vclick`

터치 이벤트 또는 `click`이 발생할 때 나타나는 이벤트이다.

`vmousecancel`

터치 이벤트 또는 `mouseleave`가 발생할 때 나타나는 이벤트이다.

Animation Events

CSS 전환에 적용되는 class를 추가하거나 삭제하는 `animationComplete` 플러그인을 지원한다.

Method & Utilities

전자정부 모바일 표준프레임워크는 \$.mobile 객체에 대한 메소드와 속성들을(properties) 제공한다.

\$.mobile.changePage (method)

프로그램 코드 상에서 페이지를 이동할 수 있도록 지원하는 메소드이다. 주로 화면전환, 페이지 로딩 등의 기능이 가능한 환경에서 링크 클릭이나 폼 전송을 할 때 내부적으로 사용된다.

Arguments

■ to (string or object, required)

- ▶ String: 절대 또는 상대 URL("about/us.html")
- ▶ Object: jQuery collection object. (\$("#about"))

■ options (object, optional)

▶ Properties:

- transition (string, default: \$.mobile.defaultPageTransition) 페이지를 나타내기 위한 화면전환(transition)이다.
- reverse (boolean, default: false) 페이지를 나타낼 때 화면전환(transition)이 일어나는 방향을 결정한다
- changeHash (boolean, default: true) 주소창의 hash가 업데이트될 필요가 있는 지 결정한다.
- role (string, default: undefined) 페이지를 화면에 출력할 때 사용되는 data-role 값이다. 각 요소에 정의된 @data-role 속성 값에 의존적이며 default 값은 'undefined' 이다.
- pageContainer (jQuery collection, default: \$.mobile.pageContainer) 페이지를 포함하고 있어야 할 요소들을 식별한다.
- type (string, default: "get") 페이지의 요청을 만들 때 method가 'get' 또는 'post' 방식인 지 식별한다. (changePage()의 to 인자(argument) 값이 URL일 때 사용된다.)
- data (object or string, default: undefined) Ajax 페이지 요청과 함께 전달될 데이터이다. (changePage()의 to 인자(argument) 값이 URL일 때 사용된다.)
- reloadPage (boolean, default: false) 페이지를 리로드(reload)하며 이미 페이지 컨테이너(page container)의 DOM에 속해있는 페이지라도 리로드를 수행한다. (changePage()의 to 인자(argument) 값이 URL일 때 사용된다.)

Examples

```
//transition to the "about us" page with a slideup transition
$.mobile.changePage( "about/us.html", { transition: "slideup" } );

//transition to the "search results" page, using data from a form with an ID of "search"
$.mobile.changePage( "searchresults.php", {
  type: "post",
  data: $("#form#search").serialize()
});

//transition to the "confirm" page with a "pop" transition without tracking it in history
$.mobile.changePage( "../alerts/confirm.html", {
  transition: "pop",
  reverse: false,
  changeHash: false
});
```

\$.mobile.loadPage (method)

외부 페이지를 로드하고, DOM에 추가한다. 이 method는 첫번째 인수로 URL이 올 때 changePage() 함수를 통해 내부적으로 호출된

다. 이 함수는 현재 활성화된 페이지에는 영향을 주지 않고, 백그라운드에서 페이지를 로드 할 때 사용된다. 이 함수는 페이지가 로딩 되고, 문서에 추가 된 후에 미리 정의된 객체를 리턴 한다.

Arguments

■ url (string, required)상대 또는 절대 URL

■ options (object, optional)

▸ Properties:

- role (string, default: undefined) 페이지를 화면에 출력할 때 사용되는 data-role 값이다. 각 요소에 정의된 @data-role 속성 값에 의존적이며 default 값은 'undefined' 이다.
- pageContainer (jQuery collection, default: \$.mobile.pageContainer) 페이지가 로드된 후 페이지를 포함하고 있어야 할 요소들을 식별한다.
- type (string, default: "get") 페이지의 요청을 만들 때 method가 'get' 또는 'post' 방식인 지 식별한다.
- data (object or string, default: undefined) Ajax 페이지 요청과 함께 전달될 데이터이다
- reloadPage (boolean, default: false) 페이지를 리로드(reload)하며 이미 페이지 컨테이너(page container)의 DOM에 속해있는 페이지라도 리로드를 수행한다.

Examples

```
//load the "about us" page into the DOM
$.mobile.loadPage( "about/us.html" );

//load a "search results" page, using data from a form with an ID of "search"
$.mobile.loadPage( "searchresults.php", {
    type: "post",
    data: $("form#search").serialize()
});
```

jqmData(), jqmRemoveData(), and jqmHasData() (method)

Arguments

jqmData(), jqmRemoveData()는 jquery core의 data와 removeData이다. Data 속성의 setting와 getting를 자동으로 만들어 준다.

Examples

\$("#div[data-role='page']") 대신 \$("#div:jqmData(role='page')") 또는 \$("#div[data-"+ \$.mobile.ns +"role='page']")를 사용 가능하다.

\$.mobile.showPageLoadingMsg ()

페이지 로딩 메시지를 보여준다.

Examples

```
//cue the page loader
$.mobile.showPageLoadingMsg();
```

\$.mobile.hidePageLoadingMsg ()

페이지 로딩 메시지를 숨긴다.

Examples

```
//cue the page loader
$.mobile.hidePageLoadingMsg();
```

\$.mobile.path.parseUrl (method)

URL을 파싱하는 함수이다. URL의 각 요소(data)에 접근을 용이하게 하며 파싱할 때 해당 값이 없으면 항목에 빈 String 값으로 채워진다. (예 protocol, host 등) 또한 권한이 없는 URL을 파싱 할 때는 protocol/scheme 콜론 뒤에 데이터가 포함된다.

Arguments

■ url (string, required) 상대 또는 절대 URL

■ Return Value

문자열 형식의 다양한 URL 컴포넌트를 포함한 객체를 반환한다. 객체의 속성은 브라우저의 location 객체를 따른다.

hash

'#' 문자(character)를 포함하는 URL 컴포넌트의 fragment 이다.

host

URL의 host와 포트 번호이다.

hostname

URL에 포함된 host 이름이다.

href

분석된 URL 원본이다.

pathname

URL에 의해 참조되는 디렉터리 또는 파일의 경로이다.

port

URL 내부에서 식별된 포트이다. 사용되는 프로토콜에 의해 default 포트를 사용할 수 있으므로 대부분의 빈 문자열로 표시된다.

protocol

'.' 문자(character)를 포함하는 URL을 지원하는 프로토콜이다.

search

'?' 문자(character)를 포함한 URL의 query 컴포넌트이다. \$.mobile.path.parseUrl() 메소드는 개발자가 접근 가능한 주요 형식의 URL 뿐만 아니라 기타 컴포넌트에 대한 접근을 지원하는 속성도 제공한다

authority

URL의 username, password, host component 이다.

directory

파일명을 제외한 pathname 의 디렉터리 컴포넌트이다.

domain

URL 의 프로토콜, 권한(authority) 에 관련된 컴포넌트이다.

filename

디렉터리를 제외한 pathname 내부의 파일명이다.

hrefNoHash

fragment(hash) 컴포넌트를 제외한 URL 의 원본이다.

hrefNoSearch

query(search) 와 fragment(hash) 를 제외한 URL 의 원본이다.

password

권한(authority) 컴포넌트 내부에 포함된 password 이다.

username

권한(authority) 컴포넌트 내부에 포함된 username 이다.

Examples

```
// Parsing the Url below results an object that is returned with the
// following properties:
//
// obj.href: http://jblas:password@mycompany.com:8080/mail/inbox?msg=1234&type=unread#msg-content
// obj.hrefNoHash: http://jblas:password@mycompany.com:8080/mail/inbox?msg=1234&type=unread
// obj.hrefNoSearch: http://jblas:password@mycompany.com:8080/mail/inbox
// obj.domain: http://jblas:password@mycompany.com:8080
// obj.protocol: http:
// obj.authority: jblas:password@mycompany.com:8080
// obj.username: jblas
// obj.password: password
// obj.host: mycompany.com:8080
// obj.hostname: mycompany.com
// obj.port: 8080
// obj.pathname: /mail/inbox
// obj.directory: /mail/
// obj.filename: inbox
// obj.search: ?msg=1234&type=unread
// obj.hash: #msg-content

var obj = $.mobile.path.parseUrl("http://jblas:password@mycompany.com:8080/mail/inbox?msg=1234");
```

\$.mobile.path.makePathAbsolute (method)

상대경로를 절대 경로로 바꿔주는 메소드 유틸리티 함수이다.

Arguments

relPath (string, required) 파일 또는 디렉터리의 상대경로
absPath (string, required) An absolute file or relative path to resolve against.

Return Value

상대경로에 대한 절대경로를 문자열로 반환한다.

Examples

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

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

\$.mobile.path.makeUrlAbsolute (method)

상대 경로 URL을 절대경로 URL로 변경하는 유틸리티 메소드 이다.

Arguments

relUrl (string, required) 상대 URL
absUrl (string, required) An absolute URL to resolve against.

Return Value

상대 URL에 대한 절대 URL을 문자열로 반환한다.

Examples

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

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

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

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

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

\$.mobile.path.isSameDomain (method)

두가지 URL을 비교해서 같은 도메인인지 판단하여 Boolean 값을 리턴한다..

Arguments

url1 (string, required) 상대 URL
url2 (string, required) 절대 URL (An absolute URL to resolve against.)

Return Value

도메인이 같으면 true를 반환하고 같지 않으면 false를 반환한다.

Examples

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

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

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

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

\$.mobile.path.isRelativeUrl (method)

URL이 상대 경로인지 판단하여 Boolean 값을 리턴한다.

Arguments

url (string, required) 상대 또는 절대 URL

Return Value

상대 URL이면 true를 반환하고 절대 URL이면 false를 반환한다.

Examples

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

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

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

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

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

// Returns: true
```

```
var isRel = $.mobile.path.isRelativeUrl("#foo");
```

\$.mobile.path.isAbsoluteUrl (method)

URL이 절대 경로인지 판단하여 Boolean 값을 리턴한다..

Arguments

url (string, required) 상대 또는 절대 URL

Return Value

절대 URL이면 true를 반환하고 상대 URL이면 false를 반환한다.

Examples

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

// Returns: false
var isAbs = $.mobile.path.isAbsoluteUrl("//foo.com/a/file.html");

// Returns: false
var isAbs = $.mobile.path.isAbsoluteUrl("/a/file.html");
// Returns: false
var isAbs = $.mobile.path.isAbsoluteUrl("file.html");

// Returns: false
var isAbs = $.mobile.path.isAbsoluteUrl(s"?a=1&b=2");

// Returns: false
var isAbs = $.mobile.path.isAbsoluteUrl("#foo");
```

\$.mobile.base (methods, properties)

일반화된 base element로 동작 하도록 한다.

\$.mobile.silentScroll (method)

스크롤 이벤트 리스너를 실행시키지 않고 인수로 지정한 Y 좌표로 스크롤한다.

Arguments

yPos (number, defaults to 0). 스크롤 할 Y 좌표 값

Examples

```
//scroll to Y 100px
$.mobile.silentScroll(100);
```

\$.mobile.addResolutionBreakpoints (method)

HTML 요소에 추가될 클래스의 최소 최대 너비에 대한 범위를 지정한다. 추가된 min/max width classes에 breakpoint를 추가한다.

Arguments

values (숫자 또는 배열 값). resolution 클래스에 추가될 숫자 또는 배열을 전달한다. 자세한 내용은 Orientation & resolution targeting 을 참조한다.

Examples

```
//add a 400px breakpoint
$.mobile.addResolutionBreakpoints(400);
//add 2 more breakpoints
$.mobile.addResolutionBreakpoints([600,800]);
```

\$.mobile.activePage (property)

화면에 보여지고 있는 현재 페이지의 참조를 리턴한다.

I Responsible Layout

모바일 기기의 다양한 스크린 크기에 맞춰서 출력하도록 설정 하는 것이며 다양한 크기의 웹사이트 지원을 위해 유동적인 CSS 적용 및 media query를 사용한다.

방향에 따른 클래스

디바이스의 가로/세로 방향에 따른 CSS를 활용 할 수 있다.

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

크기에 따른 클래스

디바이스의 너비에 따라서 설정되는 클래스이다.너비는 320,480,768,1024px 점을 가진다.

일반 클래스

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

너비가 480px 이하 일 때 적용되는 클래스

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

크기 제어 포인트

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

1200px에 Breakpoint 추가

```
$.mobile.addResolutionBreakpoints(1200);
```

2개 이상의 Breakpoint 추가

```
$.mobile.addResolutionBreakpoints([1200, 1440]);
```

Media Queries

CSS의 특정 미디어 쿼리 적용 여부를 테스트하는 기능이다. 지원하면 true, 지원하지 않으면 false가 반환된다.

screen 미디어 형식 테스트

```
$.mobile.media("screen");
```

screen 미디어 형식과 min-width 테스트

```
$.mobile.media("screen and (min-width: 480px)");
```

screen 미디어 형식과 iOS 디스플레이 여부 테스트

```
$.mobile.media("screen and (-webkit-min-device-pixel-ratio: 2)");
```

I Dynamic Page Generation

전자정부 모바일 표준 프레임워크는 기본적으로 이벤트가 발생할 때 동적으로 DOM이 생성된다. 즉 이벤트가 발생해야만 데이터를 받아 올 수 있는데 HTML형식이 아닌 JSON으로 받은 데이터 값을 받아오는 방식을 정의한다.

1. 데이터를 JSON 형태로 형성한다.

```
var categoryData = {
  animals: {
    name: "Animals",
    description: "All your favorites from aardvarks to zebras.",
    items: [
      {
        name: "Pets",
      },
      {
        name: "Farm Animals",
      },
      {
        name: "Wild Animals",
      }
    ]
  },
  colors: {
    name: "Colors",
    description: "Fresh colors from the magic rainbow.",
    items: [
      {
        name: "Blue",
      },
      {
        name: "Green",
      },
      {
        name: "Orange",
      },
      {
        name: "Purple",
      },
      {
        name: "Red",
      },
      {
        name: "Yellow",
      },
      {
        name: "Violet",
      }
    ]
  },
  vehicles: {
    name: "Vehicles",
    description: "Everything from cars to planes.",
    items: [
```

```

    {
      name: "Cars",
    },
    {
      name: "Planes",
    },
    {
      name: "Construction",
    }
  ]
}
};

```

2. JSON으로 형성된 값을 받을 **Tag**를 **body**부분에 선언한다.

내부링크

```

<h2>Select a Category Below:</h2>
<ul data-role="listview" data-inset="true">
  <li><a href="#category-items?category=animals">Animals</a></li>
  <li><a href="#category-items?category=colors">Colors</a></li>
  <li><a href="#category-items?category=vehicles">Vehicles</a></li>
</ul>

```

외부링크

```

<h2>Select a Category Below:</h2>
<ul data-role="listview" data-inset="true">
  <li><a href="/guide/api/json/animals.do">Animals</a></li>
  <li><a href="/guide/api/json/colors.do">Colors</a></li>
  <li><a href="/guide/api/json/vehicles.do">Vehicles</a></li>
</ul>

```

3. **pagebeforechange** 이벤트를 **bind** 한 후 받아온 **Data**를 처리할 스크립트를 작성한다.

```

// Listen for any attempts to call changePage().
$(document).bind( "pagebeforechange", function( e, data ) {
  // We only want to handle changePage() calls where the caller is
  // asking us to load a page by URL.
  if ( typeof data.toPage === "string" ) {
    // We are being asked to load a page by URL, but we only
    // want to handle URLs that request the data for a specific
    // category.
    var u = $.mobile.path.parseUrl( data.toPage ),
        re = /^#category-item/;
    if ( u.hash.search(re) !== -1 ) {
      // We're being asked to display the items for a specific category.
      // Call our internal method that builds the content for the category
      // on the fly based on our in-memory category data structure.
      showCategory( u, data.options );
      // Make sure to tell changePage() we've handled this call so it doesn't

```

```

        // have to do anything.
        e.preventDefault();
    }
}
});

```

```

// Load the data for a specific category, based on
// the URL passed in. Generate markup for the items in the
// category, inject it into an embedded page, and then make
// that page the current active page.
function showCategory( urlObj, options )
{
    var categoryName = urlObj.hash.replace( /.category=/, "" ),
        // Get the object that represents the category we
        // are interested in. Note, that at this point we could
        // instead fire off an ajax request to fetch the data, but
        // for the purposes of this sample, it's already in memory.
        category = categoryData[ categoryName ],
        // The pages we use to display our content are already in
        // the DOM. The id of the page we are going to write our
        // content into is specified in the hash before the '?'.
        pageSelector = urlObj.hash.replace( /\?.*$/, "" );
    if ( category ) {
        // Get the page we are going to dump our content into.
        var $page = $( pageSelector ),
            // Get the header for the page.
            $header = $page.children( ":jqmData(role=header)" ),
            // Get the content area element for the page.
            $content = $page.children( ":jqmData(role=content)" ),
            // The markup we are going to inject into the content
            // area of the page.
            markup = "<p>" + category.description + "</p><ul data-role='listview' data-inset='true'>",
            // The array of items for this category.
            citems = category.items,
            // The number of items in the category.
            numItems = citems.length;
        // Generate a list item for each item in the category
        // and add it to our markup.
        for ( var i = 0; i < numItems; i++ ) {
            markup += "<li>" + citems[i].name + "</li>";
        }
        markup += "</ul>";
        // Find the h1 element in our header and inject the name of
        // the category into it.
        $header.find( "h1" ).html( category.name );
        // Inject the category items markup into the content element.
        $content.html( markup );
        // Pages are lazily enhanced. We call page() on the page
        // element to make sure it is always enhanced before we
        // attempt to enhance the listview markup we just injected.
        // Subsequent calls to page() are ignored since a page/widget
        // can only be enhanced once.
        $page.page();
        // Enhance the listview we just injected.
        $content.find( ":jqmData(role=listview)" ).listview();
        // We don't want the data-url of the page we just modified
        // to be the url that shows up in the browser's location field,

```



```
// so set the dataUrl option to the URL for the category
// we just loaded.
options.dataUrl = urlObj.href;
// Now call changePage() and tell it to switch to
// the page we just modified.
$.mobile.changePage( $page, options );
}
}
```

예시 보기

받아온 데이터를 처리하는 방식은 크게 내부링크 방식과 외부 링크 방식 두가지가 있다.

내부링크 **Dynamic Page Example**

외부링크 **Dynamic Page Example**

I Button

버튼은 핵심적인 위젯으로서 여러 플러그인에서 두루 사용된다.

기본코드

Anchor 링크에 `data-role="button"` 속성을 추가하면 전자정부 모바일 표준프레임워크에서 제공하는 버튼의 스타일을 적용할 수 있다.

Button

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

색상변경

`data-theme` 속성 값으로 테마를 적용하여 색상을 변경한다.(총 7가지 색상 지원)

`data-theme="a"`

`data-theme="b"`

`data-theme="c"`

`data-theme="d"`

`data-theme="e"`

`data-theme="f"`

`data-theme="g"`

```
<a href="index.html" data-role="button" data-theme="c"> data-theme="c"</a>
```

형태변경

기본적인 버튼의 형태는 둥근형이며 사각형을 원할 시 기본코드에 `class="egov-btn-rectangle"` 속성을 추가한다.

Rectangle

```
<a href="#" data-role="button" class="egov-btn-rectangle">Rectangle</a>
```

버튼 그룹

여러 버튼을 하나의 그룹으로 만들려면 `controlgourp` 속성을 갖는 `<div data-role="controlgroup">` 컨테이너에 버튼을 포함시킨다. 전자정부 모바일 표준프레임워크는 기본적으로 수직으로 배열된 버튼그룹을 생성한다.

Yes

No

Maybe

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

```
<a href="#" data-role="button">No</a>
<a href="#" data-role="button">Maybe</a>
</div>
```

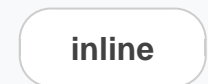
data-type="horizontal" 속성을 <div data-role="controlgroup"> 컨테이너에 추가하면 수평으로 배열된 버튼그룹을 생성할 수 있다. 수평으로 배열된 버튼그룹의 너비가 화면의 너비보다 넓은 경우 버튼그룹이 여러 줄로 보여질 수 있으므로 유의해야 한다.



```
<div data-role="controlgroup" data-type="horizontal">
```

크기변경

기본적으로 버튼은 block 스타일이 적용되어 자동으로 너비가 화면에 맞춰진다. 너비가 버튼내용에 맞춰진 버튼을 생성하려면 해당 버튼에 data-inline="true" 속성을 추가한다.



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

data-inline="true" 속성을 갖는 버튼을 연속적으로 배열하면 여러 버튼을 한 줄에 위치시킬 수 있다.



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

높이는 normal, small이 지원되며 small을 원할 시 기본코드에 class="egov-btn-small" 을 추가한다.



```
<a href="#" data-role="button" class="egov-btn-small">Small</a>
```

버튼 아이콘

data-icon 속성을 추가하여 버튼에 [아이콘](#)을 추가할 수 있다.



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

I Panel

Panel은 Header/Footer bar와 함께 페이지를 구성하는 요소이다. Panel은 여러 색상을 지원하며 겹쳐서 사용도 가능하다.

기본코드

Panel의 기본코드는 `<div class="ui-body-[a~g]">`를 사용하며 a~g는 Panel 의 색상 변경을 지원한다.

Panel

```
<div class="ui-body-c center"><h4>Panel</h4></div>
```

색상변경

색상 변경시에는 `<div class="ui-body-[a~g]">`를 변경한다.

Content A

Content B

Content C

Content D

Content E

Content F

Content G

```
<div class="ui-body-a">
  <h4>Content A</h4>
</div>
<div class="ui-body-b">
  <h4>Content B</h4>
</div>
<div class="ui-body-c">
  <h4>Content C</h4>
</div>
<div class="ui-body-d">
```

```

    <h4>Content D</h4>
</div>
<div class="ui-body-e">
    <h4>Content E</h4>
</div>
<div class="ui-body-f">
    <h4>Content F</h4>
</div>
<div class="ui-body-g">
    <h4>Content G</h4>
</div>

```

무늬 변경

Panel 의 무늬 변경은 기본코드에 class="egov-panel-pattern-a~g" 을 추가한다.

무늬 변경

```

<div class="ui-body-f egov-panel-pattern-f"><h4>무늬 변경</h4></div>

```

Internal / External Link

전자정부 모바일 표준프레임워크는 HTML의 모든 표준 링크를 지원하며 기본적으로 Ajax 방식을 이용하여 페이지를 전환한다.

유의사항

Ajax로 도메인 내부링크를 연결 후 페이지 내부링크를 호출하면 정상페이지가 아닌 이전 페이지가 출력된다.
이를 방지 하기 위해 첫 도메인 내부링크는 data-ajax="false" 또는 rel="external"를 선언하여 페이지를 호출한다.

예시보기

기본코드

HTML의 Anchor 태그를 이용하여 링크 기능을 사용할 수 있다.

페이지 내부 링크

```
<a href="#Two">페이지 내부 링크</a>
```

도메인 내부 링크

```
<a href="/guide/guide.do">도메인 내부 링크</a>
```

외부 링크

```
<a href="http://www.egovframe.go.kr">외부 링크</a>
```

Ajax를 사용하는 링크

Ajax 방식의 링크를 이용하면 다양한 화면전환 효과가 자동으로 적용된다. Ajax 방식의 링크를 사용하면 이동한 페이지 헤더 부분의 Javascript, CSS 효과가 적용되지 않을 수 있기 때문에 Javascript, CSS를 페이지를 불러들이는 Page에 정의해야한다.

■ 페이지 내부링크

data-role 속성 값이 page 인 <div data-role="page"> Tag에 id 속성을 부여하여 페이지를 구분한다. href 속성에 '[#페이지 id]' 값을 부여하여 해당 id를 갖는 페이지로의 링크가 가능하다. 또한 내부링크를 Page를 호출 하는 링크는 rel="external" or data-ajax="false"를 이용하여 페이지 내부링크 페이지를 호출해야한다.

페이지 내부링크

```
<a href="#Two" data-role="button">페이지 내부링크</a>
```

■ 도메인 내부링크

동일한 도메인 내의 페이지로 링크를 걸어 이동한다.

도메인 내부링크

```
<a href="/guide/guide.do" data-role="button">도메인 내부링크</a>
```

■ 다이얼로그 링크

data-rel="dialog" 속성을 추가하여 다이얼로그 형태로 링크를 실행한다.

다이얼로그 링크

```
<a href="/guide/guide.do" data-role="button" data-rel="dialog">다이얼로그 링크</a>
```

Ajax를 사용하지 않는 링크

Ajax 방식을 사용하지 않는 링크는 페이지 전체를 Refresh 한다. Ajax 방식을 사용하지 않으면 화면효과가 적용되지 않지만 헤더 부분의 Javascript, CSS 를 적용할 수 있다.

■ 외부 링크

외부 도메인의 사이트로 링크를 걸어 이동한다.

외부 링크

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

■ data-ajax="false" 속성 추가

data-ajax="false" 속성을 추가하여 Ajax 효과 없이 페이지를 이동한다.

data-ajax="false"

```
<a href="/guide/guide.do" data-ajax="false" data-role="button">data-ajax="false"</a>
```

■ rel="external" 속성 추가

rel="external" 속성을 추가하면 Ajax 효과 없이 페이지를 이동한다.

rel="external"

```
<a href="/guide/guide.do" rel="external" data-role="button">rel="external"</a>
```

■ target 속성 추가

target 속성을 추가하면 Ajax 효과 없이 새로운 브라우저 창으로 이동한다.

target

```
<a href="/guide/guide.do" target="home" data-role="button">target</a>
```

링크버튼

data-role="button" 속성을 추가하여 링크를 버튼 형태로 표시한다.

링크 버튼

```
<a href="/guide/guide.do" data-role="button">링크 버튼</a>
```

에러 메시지 표시

해당 페이지로 이동이 불가능한 경우 자동으로 에러 메시지를 표시한다.

에러 메시지

```
<a href="nopage.jsp" data-role="button">에러 메시지</a>
```

메일 연동 링크

href 속성 값으로 mailto:[메일주소]를 추가하면 링크를 통해 자동으로 메일기능을 실행한다.

메일링크

```
<a href="person@domain.com" data-role="button">메일링크</a>
```

href 속성 값으로 mailto:[메일주소]를 추가한후 cc="[메일주소]"를 통해 참조, bcc="[숨은 참조]", subject="[제목]", body="[내용]"을 추가 하면 다양한 형식으로 메일 연동을 할 수 있다.

다중메일링크

```
<a href="mailto:person@domain.com?cc=person2@domain.com&bcc=persion3@domain.com&subject=Hello&body=Happy%20Birthday!">다중메일링크</a>
```

통화 연동 링크

href 속성 값으로 tel:[전화번호]를 추가하면 링크를 통해 자동으로 휴대폰의 통화기능을 실행한다.

통화링크

```
<a href="tel:000000000000" data-role="button">Phone Links</a>
```


Label / Text

Label / Text 는 HTML 의 <h1>, <p> Tag 를 사용하며 CSS 정의하여 색상, 크기, 폰트를 변경 하여 사용 할 수 있다.

기본코드

■ Label

- ▶ <h1> Tag는 문서 전체의 타이틀이나 매우 중요한 부분을 나타내기 위해 사용한다.
- ▶ <h2> Tag는 문서의 보조 섹션을 나타내기 위해 사용되며 <h1> Tag 보다 비교적 중요도가 덜한 부분을 나타내기 위해 사용한다.
- ▶ 가장 높은 수준의 헤더 <h1>로 시작하고 중요도에 따라 아래와 같이 헤더의 수준을 낮춰서 사용하는 것을 권장한다.

H1 Heading

H2 Heading

H3 Heading

H4 Heading

H5 Heading

H6 Heading

```
<h1>H1 Heading</h1>
<h2>H2 Heading</h2>
<h3>H3 Heading</h3>
<h4>H4 Heading</h4>
<h5>H5 Heading</h5>
<h6>H6 Heading</h6>
```

■ Text

- ▶ HTML의 <p>, <div> Tag 를 사용하며 CSS 를 정의하여 색상, 폰트, 크기를 변경 한다. 스타일시트를 이용하여 텍스트의 배열을 다르게 설정해 줄 수 있다.

left

center

right

```
<div class="egov-align-left">left</div>
<div class="egov-align-center">center</div>
<div class="egov-align-right">right</div>
```

스타일 시트를 이용하여 폰트를 다르게 설정해 줄 수 있다.

This paragraph is in helvetica.

This paragraph is in verdana.
This paragraph is in tahoma.

```
<div class="egov-font-helvetica">This paragraph is in helvetica.</div>  
<div class="egov-font-verdana">This paragraph is in verdana.</div>  
<div class="egov-font-tahoma">This paragraph is in tahoma.</div>
```

스타일 시트를 이용하여 텍스트의 크기를 다르게 설정해 줄 수 있다.

This paragraph is in size 30px.
This paragraph is in size 25px.
This paragraph is in size 15px.

```
<div class="egov-font-size30">This paragraph is in size 30px.</div>  
<div class="egov-font-size25">This paragraph is in size 25px.</div>  
<div class="egov-font-size30">This paragraph is in size 15px.</div>
```

스타일 시트를 이용하여 텍스트의 색상을 다르게 설정해 줄 수 있다.

This paragraph is in green.
This paragraph is in red.
This paragraph is in blue.

```
<div class="egov-color-green">This paragraph is in green.</div>  
<div class="egov-color-red">This paragraph is in red.</div>  
<div class="egov-color-blue">This paragraph is in blue.</div>
```

I Tabs

탭은 버튼이 수평으로 나열된 형태로 모바일 환경에 최적화된 내비게이션 UI를 제공한다. Navbar(Tab)는 Header와 Footer 영역에 사용되며 1~5개 까지 배치 할 수 있고 옵션을 통해 icon 표시가 가능 하다.

기본코드

data-role="navbar" 를 이용하여 내비게이션 기능을 사용하며 ui-btn-active 클래스 값으로 활성화된 탭을 나타낼 수 있다.

eGovFrame

OneTwoThree

eGovFrame

- [One](#)
- [Two](#)
- [Three](#)

navbar 배치

 Tag를 사용하여(1~5개) Navbar 의 배치를 설정 할 수 있다. Navbar 항목에 Tag를 1개 사용할 경우 브라우저 화면의 100% 너비를 사용한다.

eGovFrame

One

- [One](#)

Navbar 항목에 Tag를 2개 사용할 경우 브라우저 화면의 100%를 1/2 씩 나누어 배치 한다.

eGovFrame

OneTwo

- [One](#)
- [Two](#)

Navbar 항목에 Tag를 3개 사용할 경우 브라우저 화면의 100%를 1/3 씩 나누어 배치 한다.

eGovFrame

One

Two

Three

<div data-role="navbar">

 One
 Two
 Three

</div>

Navbar 항목에 Tag를 4개 사용할 경우 브라우저 화면의 100%를 1/4 씩 나누어 배치 한다.

eGovFrame

One

Two

Three

Four

<div data-role="navbar">

 One
 Two
 Three
 Four

</div>

Navbar 항목에 Tag를 5개 사용할 경우 브라우저 화면의 100%를 1/5 씩 나누어 배치 한다.

eGovFrame

One

Two

Three

Four

Five

<div data-role="navbar">

 One
 Two
 Three
 Four
 Five

</div>

Navbar 항목에 Tag를 5개 이상 사용할 경우 한 행에 2개열로 구성된 단순한 다중행 형태로 배치 된다.

eGovFrame

One

Two

Three

Four

Five

Six

Seven	Eight
Nine	Ten

```
<div data-role="navbar">
  <ul>
    <li><a href="#" class="ui-btn-active">One</a></li>
    <li><a href="#">Two</a></li>
    <li><a href="#">Three</a></li>
    <li><a href="#">Four</a></li>
    <li><a href="#">Five</a></li>
    <li><a href="#">Six</a></li>
    <li><a href="#">Seven</a></li>
    <li><a href="#">Eight</a></li>
    <li><a href="#">Nine</a></li>
    <li><a href="#">Ten</a></li>
  </ul>
</div>
```

형태변경 (둥근 탭)

Navbar 내부의 Anchor Tag에 class="egov-btn-corner-tl" 속성을 추가하여 왼쪽으로 둥근 탭을 생성할 수 있다. class="egov-btn-corner-tr" 속성을 추가하면 오른쪽으로 둥근 탭을 생성할 수 있다.

eGovFrame		
Left	Center	Right

```
<div data-role="header" data-theme="b">
  <h1>eGovFrame</h1>
  <div data-role="navbar">
    <ul>
      <li><a href="#" class="egov-btn-corner-tl">Left</a></li>
      <li><a href="#">Center</a></li>
      <li><a href="#" class="egov-btn-corner-tr">Right</a></li>
    </ul>
  </div>
</div>
```

문서 간 이동

문서 간 이동은 활성화 tab의 class="ui-btn-active" 속성을 설정해 주어 사용 가능하다.

[문서 간 이동 샘플 보기](#)

```
<div data-role="header" data-position="inline" data-theme="b">
  <h1>Mobile FrameWork UX Components</h1>
  <div data-role="navbar">
    <ul>
      <li><a href="#" class="ui-btn-active">One</a></li>
      <li><a href="#">Two</a></li>
      <li><a href="#">Three</a></li>
    </ul>
  </div>
</div>
```

문서 내 이동

문서 내 이동은 data-role="tabs" 설정을 통하여 사용 가능하다.

[문서 내 이동 샘플 보기](#)

```
<div data-role="header" data-position="inline" data-theme="a">
  <h1 class="main">eGovFrame</h1>
  <div data-role="tabs">
    <ul>
      <li><a href="#tab-1" >Tab 1</a></li>
      <li><a href="#tab-2" >Tab 2</a></li>
      <li><a href="#tab-3" >Tab 3</a></li>
    </ul>
  </div>
</div>
<div data-role="content">
  <div id="tab-1">
    <h1><font color="red">Here is the first tab</font></h1><br/><br/>
  </div>
  <div id="tab-2">
    <h1><font color="blue">Here is the second tab</font></h1><br/><br/>
  </div>
  <div id="tab-3">
    <h1><font color="green">Here is the third tab</font></h1><br/><br/>
  </div>
</div>
```

Icon 추가

data-icon="icon" 속성을 사용하여 Tabs에 icon 을 추가 할 수 있다.



```
<div data-role="navbar">
  <ul>
    <li><a href="#" data-icon="grid">Summary</a></li>
    <li><a href="#" data-icon="star" class="ui-btn-active">Favs</a></li>
    <li><a href="#" data-icon="gear">Setup</a></li>
  </ul>
</div>
```

I Form

HTML의 모든 폼 요소를 모바일 환경에 최적화된 UI로 제공한다. 폼을 사용할 때는 여러 페이지가 하나의 DOM에 속할 수 있으므로 id 속성을 사이트 범위에서 유일한 값으로 지정해야 한다.

기본코드

전자정부 모바일 표준프레임워크를 이용하여 폼을 생성하면 레이블 및 폼 구성요소가 화면의 너비에 맞게 자동으로 배열된다. 폼 요소 배열의 구성을 향상시키려면 `<div data-role="fieldcontain">` Tag 내부에 레이블 및 폼 요소를 포함시킨다.

Text :

Password :

Submit

```
<form id="form-1" action="#" method="get">
  <div data-role="fieldcontain">
    <label for="text">Text :</label>
    <input type="text" id="text"/>
  </div>
  <div data-role="fieldcontain">
    <label for="password">Password :</label>
    <input type="password" id="password"/>
  </div>
  <div data-role="fieldcontain">
    <button id="submit" type="submit">Submit</button>
  </div>
</form>
```

레이블과 폼 구성요소

화면의 너비가 좁을 경우 레이블이 폼 구성요소의 상단에 위치하게 되고 화면의 너비가 넓을 경우에는 레이블과 폼 구성요소가 한 줄에 나타나게 된다. 이때 레이블과 폼 요소가 짝을 이룰 수 있도록 폼 구성요소의 id 속성과 `<label>` Tag의 for 속성 값을 일치시켜야 한다.

Text :

Password :

Text :

Password :

Submit

```
<form id="form-2" action="#" method="get">
```

```

<label for="text1">Text : </label>
<input type="text" id="text1"/>
<label for="password1">Password :</label>
<input type="password" id="password1"/>
<label for="text2">Text : </label>
<input type="text" id="text2"/>
<label for="password2">Password :</label>
<input type="password" id="password2"/>
<button data-theme="b" id="submit" type="submit">Submit</button>
</form>


```

리스트 형태의 폼

<form> Tag 내부에 리스트 태그(ul, ol)를 두고 Tag에 폼 요소를 배치하여 리스트 형태로 구분된 폼을 생성할 수 있다.

Text :

Password :

Slider: 

Submit

```

<form id="form-4" action="#" method="get">
  <ul data-role="listview" data-inset="true">
    <li data-role="fieldcontain">
      <label for="text">Text :</label>
      <input type="text" id="text"/>
    </li>
    <li data-role="fieldcontain">
      <label for="password">Password :</label>
      <input type="password" id="password"/>
    </li>
    <li data-role="fieldcontain">
      <label for="slider">Slider:</label>
      <input type="range" id="slider" value="0" min="0" max="100" />
    </li>
    <li data-role="fieldcontain">
      <button data-theme="b" id="submit" type="submit">Submit</button>
    </li>
  </ul>
</form>

```

향상된 Input Type

HTML5 의 다양한 Input Type 을 사용하면 모바일 기기에서 자동으로 해당 Input Type 에 맞는 입력 도구를 제공한다.

Number Input :

Email Input :

URL Input :

TEL Input :

폼 구성요소

폼 형식으로 주어진 레이아웃에 Tag를 추가하여 폼 구성요소를 추가한다. 폼 구성요소의 종류는 Text inputs, Search inputs, Sliders, Switches, Radio buttons, Check boxes, Selectors 가 있다.

Text :

Password :

Textarea:

Search Input:

Switch :

☐ Off

Slider:

Radio Button:

<input checked="" type="radio"/>	선택 1
<input type="radio"/>	선택 2
<input type="radio"/>	선택 3
<input type="radio"/>	선택 4

Check Box :

<input type="checkbox"/>	동의합니다.
--------------------------	--------

Check Box :

<input type="checkbox"/>	다중선택 1
<input type="checkbox"/>	다중선택 2

다중선택 3

다중선택 4

Check Box :

아침

점심

저녁

Radio Button:

시작

끝

기타

Select Menu:

메뉴 선택 1

▼

Submit

폼 버튼

<button> Tag 를 사용하거나 <input> Tag 에 "button", "submit", "image" 속성을 추가하여 폼 버튼을 생성할 수 있으며 button의 경우 onclick으로 submit 이벤트를 걸어준다. button의 submit이벤트는 Ajax가 false인 방법으로 전송된다. 또한 이들은 Anchor 태그 형식의 버튼과 마찬가지로 data-inline, data-icon 등의 버튼 속성을 적용할 수 있다.

Button Tag

Submit Input

Image Input

Button Input

```
<button>Button Tag</button>
<input type="submit" value="Submit Input" />
<input type="image" value="Image Input" />
<input type="button" value="Button Input" onClick="javascript:btnSubmit();" />
```

HTML 스타일의 폼

폼의 각 요소에 data-role="none"으로 속성 값을 지정하면 기존 HTML 에서 제공하는 고전적인 형태의 폼을 생성할 수 있다.

Text :

Switch : Off

Radio Button: 선택 1 선택 2 선택 3

Check Box : EMail SMS

Cancel Submit

I Menu

모바일 Page의 메뉴 기능은 Dialog, Grid, List, Collapsible 의 기능을 활용하여 구현할 수 있다.

Menu Type

Dialog, Grid Menu



Dialog 형식

링크에 data-rel="dialog" 및 data-transition="slide, slidedown, slideup, fade, flip" 속성을 추가하여 다이얼 로그의 효과적인 표출 효과를 사용 할 수 있다.

※ dialog 형식의 메뉴는 [UX Dialog Component](#) 유의 사항을 참고한다.

Menu	
표준프레임워크 개요	➤
다운로드	➤
가이드	➤
자료실	➤
개발자 지원	➤
오픈 커뮤니티	➤

```
<ul data-role="listview" data-inset="true">
  <li data-role="list-divider">Menu</li>
  <li><a href="#menu1" data-rel="dialog" data-transition="slide">표준프레임워크 개요</a></li>
  <li><a href="#menu2" data-rel="dialog" data-transition="slidedown">다운로드</a></li>
  <li><a href="#menu3" data-rel="dialog" data-transition="slideup">가이드</a></li>
  <li><a href="#menu4" data-rel="dialog" data-transition="pop">자료실</a></li>
  <li><a href="#menu5" data-rel="dialog" data-transition="fade">개발자 지원</a></li>
  <li><a href="#menu6" data-rel="dialog" data-transition="flip">오픈 커뮤니티</a></li>
</ul>
```

Grid 형식

비슷한 레벨의 콘텐츠를 아이콘화 하여 여러 개 배치하여 사용하며, 메뉴 정렬 순서는 좌측에서 우측, 위에서 아래 방향으로 배치한다.

개요	다운로드	가이드
자료실	개발자	커뮤니티

```
<div class="ui-grid-b">
  <div class="ui-block-a"><a href="" data-role="button" data-theme="a">개요</a></div>
  <div class="ui-block-b"><a href="" data-role="button" data-theme="b">다운로드</a></div>
  <div class="ui-block-c"><a href="" data-role="button" data-theme="c">가이드</a></div>
  <div class="ui-block-a"><a href="" data-role="button" data-theme="d">자료실</a></div>
  <div class="ui-block-b"><a href="" data-role="button" data-theme="e">개발자</a></div>
  <div class="ui-block-c"><a href="" data-role="button" data-theme="g">커뮤니티</a></div>
</div>
```

I Progress Dialog / Bar

Progress Dialog 와 Bar 는 모바일 Page 전환 시 진행 상태를 표시해 주는 역할을 한다.

Progress Type :

Progress Dialog



Theme :

data-theme="a"



Progress dialog

Progress Bar 는 Ajax 통신을 이용할 경우 호출 되고 설정을 통해 Default 로 사용 가능하며, 수동으로 호출 하는 방법은 아래의 샘플과 같다.

페이지 이동 시 : Progress dialog

Loading

Stop

```
<a href="/guide/guide.do" data-role="button">페이지 이동 시 : Progress dialog</a>
<a href="javascript:$.mobile.showPageLoadingMsg('a');" data-role="button">Loading</a>
<a href="javascript:$.mobile.hidePageLoadingMsg('a');" data-role="button">Stop</a>
```

* Progress Dialog / Bar 의 메시지를 변경 하기 위해서는 아래와 같은 설정이 필요하다.

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

<script type="text/javascript" src="/js/egovframework/mb/cmm/jquery.mobile-1.0b3.min.js"></script>
```

* 페이지 이동시 Progress Dialog / Bar 의 테마를 변경 하기 위해서는 아래와 같은 설정이 필요하다.

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

<script type="text/javascript" src="/js/egovframework/mb/cmm/jquery.mobile-1.0b3.min.js"></script>
```

I Dialog

기본 다이얼로그는 Anchor 링크가 있는 곳에 다이얼로그를 나타 낼수있으며 다이얼로그 속성이 지정되면 대화상자 효과를 나타내기 위하여 주위 여백이 어두운 배경으로 나타난다.

유의사항

Ajax로 도메인 내부링크를 연결 후 Dialog를 호출하면 정상페이지가 아닌 이전 페이지가 출력된다.
이를 방지 하기 위해 첫 도메인 내부링크는 data-ajax="false" 또는 rel="external"를 선언하여 페이지를 호출한다.

예시보기

기본 코드

다이얼로그를 새 창으로 전환 할 경우 팝업 효과 등으로 나타 낼 수 있으며 Anchor Tag에 data-rel="dialog"를 추가 하면 다이얼로그 효과가 나타난다.

Open dialog

```
<a href="/guide/components/back.do" data-rel="dialog" data-role="button">Open dialog</a>
```

효과 변경

기본 다이얼로그 효과는 팝업 효과 이며, 다른 전환 효과를 나타낼 경우 기본 코드에 data-transition="[slide, slideup, slidedown, pop, fade, flip]"를 넣어 준다.

slide

slideup

slidedown

pop

fade

flip

```
<a href="/guide/components/dialog/back.do" data-role="button" data-rel="dialog" data-transition="slide">slide</a>  
<a href="/guide/components/dialog/back.do" data-role="button" data-rel="dialog" data-transition="slideup">slideup</a>  
<a href="/guide/components/dialog/back.do" data-role="button" data-rel="dialog" data-transition="slidedown">slidedown</a>  
<a href="/guide/components/dialog/back.do" data-role="button" data-rel="dialog" data-transition="pop">pop</a>  
<a href="/guide/components/dialog/back.do" data-role="button" data-rel="dialog" data-transition="fade">fade</a>  
<a href="/guide/components/dialog/back.do" data-role="button" data-rel="dialog" data-transition="flip">flip </a>
```

기타 Dialog는 자바 스크립트를 이용하여 Alert, Confirm 등의 창을 띄운다. 아래 스크립트 코드를 EgovMobile.js와 jquery.mobile-1.0b3.min.js 사이에 선언하여 이벤트를 걸어 사용한다. 각 Dialog는 Dialog의 id값을 참조하여 스크립트가 실행된다.

Action sheet

여러 선택창을 Dialog로 선택할 경우 사용되며 'jActionSheet' 함수를 지정하여 지정하여 사용한다.

Action Sheet

you entered :

```
$("#actionsheet_dialog").click( function() {  
    var btmlItem = [{id : 'button1', value: "커피"},  
                    {id : 'button2', value: "사이다"},  
                    {id : 'button3', value: "맥주"}];  
    jActionSheet('Can you make a choice?', 'Action sheet Dialog', 'a', btmlItem , function(r) {  
        $('#actionsheetOut').text(r);  
    });  
});
```

Overlay

이벤트 링크에 대한 부가 설명 등을 할 때 사용되며 해당 'jOverlay' 함수를 사용하며 이벤트가 걸리면 사라진다.

Overlay

```
$("#overlay_dialog").click( function() {  
    jOverlay('This is a Overbox', 'OverLay', 'a');  
});
```

Alert

알림 창이며 'jAlert' 함수를 이용하여 확인 버튼을 누르면 창이 사라진다.

Show Alert

```
$("#alert_dialog").click( function() {  
    jAlert('This is a custom alert box', 'Alert Dialog', 'a');  
});
```

confirm

확인 창이며 jConfirm 함수를 이용하여 확인 또는 닫기 버튼을 누르면 창이 사라진다..

Show Confirm

```
$("#confirm_dialog").click( function() {  
    jConfirm('Can you confirm this?', 'Confirmation Dialog', 'a', function(r) {
```



```
});});
```

prompt

문자를 입력할 수 있는 창이 나타나며 'jPrompt' 함수를 이용, 확인 버튼을 누르면 문자가 입력된다.

Show Prompt

you entered :

```
$("#prompt_dialog").click( function() {  
    jPrompt('Type something:', 'Prefilled value', 'Prompt Dialog', 'a', function(r) {  
        $('#promptOut').text(r);  
    });  
});
```

색상 변경

기타 Dialog는 색상을 변경 할 수 있는 문자를 입력 받을 수 있으며 세번째 파라미터로 색상을 지정 할 수 있다. 또한 색상 파라미터는 총 7가지 색상을 지원하며 null 값이 입력 되는 경우 상위 테마를 적용한다.

Overlay 색상 변경

```
jOverlay('e 데마 적용', 'EgovFrame', 'e');
```

Grid View

Grid View는 UX Component를 같은 넓이, 같은 행으로 나타 낼 때 사용되며 Grid View 안에 컴포넌트를 넣어 나란하게 배열 할 수 있다.

기본코드

Grid 속성으로 칼럼(column) 수를 지정하고 Block 속성으로 내부의 셀을 지정한다. class 속성 값을 ui-grid-[a~d] 갖도록 하여 격자 형태의 레이아웃을 구성하며 ui-grid-[a~d] 는 각각 [2~5]개의 칼럼을 제공한다. 기본적으로 각 칼럼의 너비는 동일하며 class 속성 값을 ui-block-[a~e] 으로 설정하여 내부 셀을 추가 한다.

I'm Block A and text inside will wrap

I'm Block B and text inside will wrap

```
<div class="ui-grid-a">
  <div class="ui-block-a"><strong>I'm Block A</strong> and text inside will wrap</div>
  <div class="ui-block-b"><strong>I'm Block B</strong> and text inside will wrap</div>
</div>
```

한 줄 Grid(동일 넓이)

기본 코드에 class="ui-grid-[a~d]"를 추가한다.

Grid Layout 에서 칼럼의 너비는 class 속성 값을 ui-grid-[a~d] 값에 따라 전체 Grid의 크기를 N등분한 길이이다.

■ ui-grid-a : 2등분

ui-block-a

ui-block-b

```
<div class="ui-grid-a">
  <div class="ui-block-a"><div class="ui-body-e" align="center"><h3>ui-block-a</h3></div></div>
  <div class="ui-block-b"><div class="ui-body-e" align="center"><h3>ui-block-b</h3></div></div>
</div>
```

■ ui-grid-b : 3등분

ui-block-a

ui-block-b

ui-block-c

```
<div class="ui-grid-b">
  <div class="ui-block-a"><div class="ui-body-e" align="center"><h3>ui-block-a</h3></div></div>
  <div class="ui-block-b"><div class="ui-body-e" align="center"><h3>ui-block-b</h3></div></div>
  <div class="ui-block-c"><div class="ui-body-e" align="center"><h3>ui-block-c</h3></div></div>
</div>
```

■ ui-grid-c : 4등분

ui-block-a

ui-block-b

ui-block-c

ui-block-d

```
<div class="ui-grid-c">
  <div class="ui-block-a"><div class="ui-body-e" align="center"><h3>ui-block-a</h3></div></div>
  <div class="ui-block-b"><div class="ui-body-e" align="center"><h3>ui-block-b</h3></div></div>
```

```
<div class="ui-block-c"><div class="ui-body-e" align="center"><h3>ui-block-c</h3></div></div>
<div class="ui-block-d"><div class="ui-body-e" align="center"><h3>ui-block-d</h3></div></div>
</div>
```

■ ui-grid-d : 5등분

ui-block-a	ui-block-b	ui-block-c	ui-block-d	ui-block-e
-------------------	-------------------	-------------------	-------------------	-------------------

```
<div class="ui-grid-d">
  <div class="ui-block-a"><div class="ui-body-e" align="center"><h3>ui-block-a</h3></div></div>
  <div class="ui-block-b"><div class="ui-body-e" align="center"><h3>ui-block-b</h3></div></div>
  <div class="ui-block-c"><div class="ui-body-e" align="center"><h3>ui-block-c</h3></div></div>
  <div class="ui-block-d"><div class="ui-body-e" align="center"><h3>ui-block-d</h3></div></div>
  <div class="ui-block-e"><div class="ui-body-e" align="center"><h3>ui-block-e</h3></div></div>
</div>
```

다중 행 Grid(동일넓이)

<div class="ui-grid-[a~d]"> Tag 에 <div class="ui-block-[a~d]"> Tag를 반복적으로 추가하여 여러 개의 행을 생성할 수 있다.

ui-block-a	ui-block-b	ui-block-c
ui-block-a	ui-block-b	ui-block-c

```
<div class="ui-grid-b">
  <div class="ui-block-a"><div class="ui-body-e" align="center"><h3>ui-block-a</h3></div></div>
  <div class="ui-block-b"><div class="ui-body-e" align="center"><h3>ui-block-b</h3></div></div>
  <div class="ui-block-c"><div class="ui-body-e" align="center"><h3>ui-block-c</h3></div></div>
  <div class="ui-block-a"><div class="ui-body-e" align="center"><h3>ui-block-a</h3></div></div>
  <div class="ui-block-b"><div class="ui-body-e" align="center"><h3>ui-block-b</h3></div></div>
  <div class="ui-block-c"><div class="ui-body-e" align="center"><h3>ui-block-c</h3></div></div>
</div>
```

한 줄 Grid(다른 넓이)

<div class="egov-grid">내 class="egov-wid[1~11]"의 class 속성을 사용하여 총 가로 길이가 egov-wid12로 표현되며 각 grid의 크기를 다르게 줄 수 있다.

* 총 grid의 합이 12가 되어야 한다.

wid3	wid9
-------------	-------------

```
<div class="egov-grid">
  <div class="egov-wid3"><div class="ui-bar-c" align="center"><h3>wid3</h3></div></div>
  <div class="egov-wid9"><div class="ui-bar-c" align="center"><h3>wid9</h3></div></div>
</div>
```

--	--

wid6	wid6
------	------

```
<div class="egov-grid">
  <div class="egov-wid6"><div class="ui-bar-c" align="center"><h3>wid6</h3></div></div>
  <div class="egov-wid6"><div class="ui-bar-c" align="center"><h3>wid6</h3></div></div>
</div>
```

다중 행 Grid(다른넓이)

<div class="egov-wid[1~11]"> Tag에 <div class="egov-wid[1~11]"> Tag를 반복적으로 추가하여 여러 개의 Row 생성이 가능하다.

wid2	wid10		
wid3		wid9	
wid10			wid2

```
<div class="egov-grid">
  <div class="egov-wid2"><div class="ui-bar-c" align="center"><h3>egov-wid2</h3></div></div>
  <div class="egov-wid10"><div class="ui-bar-c" align="center"><h3>egov-wid10</h3></div></div>
  <div class="egov-wid3"><div class="ui-bar-c" align="center"><h3>egov-wid3</h3></div></div>
  <div class="egov-wid9"><div class="ui-bar-c" align="center"><h3>egov-wid9</h3></div></div>
  <div class="egov-wid10"><div class="ui-bar-c" align="center"><h3>egov-wid10</h3></div></div>
  <div class="egov-wid2"><div class="ui-bar-c" align="center"><h3>egov-wid2</h3></div></div>
</div>
```

응용

Grid 선언 클래스 내 컴포넌트 들을 넣으면 해당 컴포넌트가 그 크기에 맞게 나타난다.

block1	block2
--------	--------

```
<div class="ui-grid-a">
  <div class="ui-block-a"><button type="submit" data-theme="c">block1</button></div>
  <div class="ui-block-b"><button type="submit" data-theme="b">block2</button></div>
</div>
```

wid5	wid7
------	------

```
<div class="egov-grid">
  <div class="egov-wid7"><button type="submit" data-theme="a">button1</button></div>
  <div class="egov-wid5"><button type="submit" data-theme="b">button2</button></div>
</div>
```

List

전자정부 모바일 표준프레임워크를 통해 생성된 리스트는 모바일 환경에 최적화된 형태로 제공된다. 리스트의 너비는 화면의 너비에 자동으로 맞춰지며 리스트 아이템 오른쪽에 arrow-r 아이콘이 적용된다. 리스트는 데이터의 정렬, 내비게이션(링크), 결과 목록, 데이터 입력폼 등 폭 넓은 기능을 수행하므로 전자정부 모바일 표준프레임워크에서는 다양한 종류의 리스트를 지원하고 있다.

기본코드

리스트 Tag(ol, ul)에 data-role="listview" 속성을 추가하면 전자정부 모바일 표준프레임워크에서 제공하는 리스트를 생성할 수 있다. 페이지 내용의 레이아웃에 맞춰 리스트를 삽입하려면 data-inset="true" 속성을 리스트 Tag에 추가한다. data-inset 속성을 true 로 지정하면 리스트에 여백과 둥근 모서리가 적용된다.

List Item 1

List Item 2

List Item 3

```
<ul data-role="listview" data-inset="true">
  <li>List Item 1</li>
  <li>List Item 2</li>
  <li>List Item 3</li>
</ul>
```

리스트 링크

Tag에 Anchor 링크를 추가하면 내비게이션 기능을 수행하는 리스트 링크를 생성할 수 있다.

Link to Home



Link to egovframe



```
<ul data-role="listview" data-inset="true">
  <li><a href="/guide/guide.do">Link to Home</a></li>
  <li><a href="http://www.egovframe.go.kr">Link to egovFrame</a></li>
</ul>
```

확장된 기능의 리스트

Nested List



Numbered List



Split Button	➤
List Divider	➤
Count Bubble	➤
Thumbnail	➤
List Icon	➤
Content Formatting	➤
Search Filter Bar	➤

리스트 업데이트

refresh() 메소드를 이용하여 리스트의 스타일 및 내용을 동적으로 업데이트할 수 있다.

refresh

refresh 버튼을 눌러 주세요.

```
$('#findList').listview('refresh');
```

I Date picker

Date picker는 팝업 형태로 나타나며 날짜를 선택할 수 있도록 기능을 제공한다.

유의사항

Ajax로 도메인 내부링크를 연결 후 팝업형태의 picker를 출력시 picker가 닫히지 않는다.(width가 480일 경우 자동적으로 팝업형태로 출력) 이를 방지 하기 위해 첫 도메인 내부링크는 data-ajax="false" 또는 rel="external"를 선언하여 페이지를 호출하거나 팝업형태의 picker가 출력되지 않도록 한다.

예시 보기

기본코드

Date Picker를 사용 하기 위해서는 아래의 자바 스크립트 적용 순서를 아래와 같이 지정해 줘야한다.

jquery.mobile-1.0b3.min.js 위에 적용

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

스크립트 가장 아래에 적용

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

안드로이드형태

안드로이드 형태는 상/하 버튼을 눌러 날짜를 선택 할 수 있다.

Some Date

```
<div data-role="fieldcontain">
<label for="defandroid">Some Date</label>
<input name="defandroid" type="date" data-role="datebox" id="defandroid"/>
</div>
```

달력 형태

안드로이드 형태에 <input name="mydate"> Tag에 data-options='{ "mode": "calbox" }'를 추가한다.

Some Date



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

Time picker

Time picker는 팝업 형태로 나타나며 시간을 선택 할 수 있도록 한다.

24 시간 단위

안드로이드 형태에 <input name="mydate"> Tag에 data-options="{\"mode\": \"timebox\"}\"를 추가한다.

Some Time
(24hour)



```
<div data-role="fieldcontain">
  <label for="mydate">Some Time (24hour)</label>
  <input name="mydate" id="mydate" type="date" data-role="datebox" data-options="{\"mode\": \"timebox\"}\">
</div>
```

12 시간 단위

안드로이드 형태에 <input name="mydate"> Tag에 data-options="{\"mode\": \"timebox\", \"timeFormat\": 12}\"'를 추가한다.

Some Time
(12hour)



```
<div data-role="fieldcontain">
  <label for="mydate">Some Time (24hour)</label>
  <input name="mydate" id="mydate" type="date" data-role="datebox" data-options="{\"mode\": \"timebox\", \"timeFormat\": 12}\">
</div>
```

색상 변경

Date/Time picker 태그 내부에 data-options="{\"page\": \"[a~g]\", \"button\": \"[a~g]\"}\"을 추가 한다. data-options 내부에 정의된 page는 picker의 전체 페이지 색상 및 확인버튼의 색상을 나타내며 button은 변경버튼(+, -)버튼의 색상을 의미한다. 또한 달력 형식의 Date picker는 data-options="{\"button\": \"[a~g]\"}\"만 정의해도 무방하다. 색상 변경은 [theme type](#) 페이지를 참조 한다.

Some Date



```
<div data-role="fieldcontain">
  <label for="mydate">Some Time (24hour)</label>
  <input name="mydate" id="mydate" type="date" data-role="datebox" data-options='{ "page": "a", "button": "f"}'>
</div>
```

Some Date



```
<div data-role="fieldcontain">
  <label for="mydate">Some Time (24hour)</label>
  <input name="mydate" id="mydate" type="date" data-role="datebox" data-options='{ "mode": "calbox", "button": "e"}'>
</div>
```

Check Box

체크 박스는 하나 이상의 항목을 선택 할 수 있는 기능을 제공한다.

기본코드

체크 박스를 세로로 정렬 하기 위해서는 <fieldset data-role="controlgroup">으로 체크 박스를 감싸 주어야 한다. 또한 기본으로 체크된 효과를 나타내고 싶을 때는 <input type="checkbox"> Tag에 checked="checked"를 추가 한다.

취미 선택 :

☐ 음악감상

☒ 영화감상

```
<fieldset data-role="controlgroup">
  <legend>취미 선택 :</legend>
  <input type="checkbox" name="checkbox-1" id="checkbox-1" class="custom" />
  <label for="checkbox-1">음악감상</label>
  <input type="checkbox" name="checkbox-2" id="checkbox-2" class="custom" checked="checked"/>
  <label for="checkbox-2">영화감상</label>
</fieldset>
```

형태 변경

체크 박스는 가로 정렬도 가능하며 그룹에서 하나를 선택 할 수 있다. 가로 형으로 그룹화할 때는 기본코드의 <fieldset data-role="controlgroup"> Tag에 data-type="horizontal"을 추가 한다.

취미 선택 :

음악감상

영화감상

```
<div data-role="fieldcontain">
  <fieldset data-role="controlgroup" data-type="horizontal">
    <legend>취미 선택 :</legend>
    <input type="checkbox" name="checkbox-1" id="checkbox-1" class="custom" />
    <label for="checkbox-1">음악감상</label>
    <input type="checkbox" name="checkbox-2" id="checkbox-2" class="custom" />
    <label for="checkbox-2">영화감상</label>
  </fieldset>
</div>
```

새로고침

Check Box는 새로고침 기능이 제공된다. 자바 스크립트에 아래 코드를 추가 한 후 이벤트로 연결 해주면 체크된 박스가 해제된다.

취미 선택 :

☒ 음악감상

☐ 영화감상

refresh

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

Radio Button

라디오 버튼은 여러개의 항목에서 하나를 선택 할 수 있는 기능을 제공한다.

기본코드

라디오 버튼을 사용 할 경우 `<input type="radio">`의 name 속성이 동일 해야하며 세로로 정렬 하기 위해서는 `<fieldset data-role="controlgroup">`으로 Radio 버튼을 감싸 주어야 한다. 또한 기본으로 체크된 효과를 나타내고 싶을 때는 `<input type="radio">` Tag에 `checked="checked"`를 추가 한다.

동의 여부:

<input checked="" type="radio"/> Yes
<input type="radio"/> No

```
<div data-role="fieldcontain">
  <fieldset data-role="controlgroup">
    <legend>동의 여부: </legend>
    <input type="radio" name="radio-1" id="radio-1" value="choice-1" checked="checked" />
    <label for="radio-1">Yes</label>
    <input type="radio" name="radio-1" id="radio-2" value="choice-2" />
    <label for="radio-2">No</label>
  </fieldset>
</div>
```

형태 변경

라디오 버튼은 가로 정렬도 가능하며 그룹에서 하나를 선택 할 수 있다. 가로 형으로 그룹화할 때는 기본코드의 `<fieldset data-role="controlgroup">` Tag에 `data-type="horizontal"`을 추가 한다.

동의 여부:

Yes	No
-----	----

```
<div data-role="fieldcontain">
  <fieldset data-role="controlgroup" data-type="horizontal">
    <legend>동의 여부: </legend>
    <input type="radio" name="radio-2" id="radio-1" value="choice-1" checked="checked" />
    <label for="radio-1">&gt;Yes</label>
    <input type="radio" name="radio-2" id="radio-2" value="choice-2" />
    <label for="radio-2">No</label>
  </fieldset>
</div>
```

새로고침

Radio button은 새로고침 기능이 제공된다. 자바 스크립트에 아래 코드를 추가 한 후 이벤트로 연결 해주면 체크된 버튼이 해제된다.

동의 여부 :

<input checked="" type="radio"/> Yes
<input type="radio"/>

☐ No

refresh

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

ICon

모바일 어플리케이션에 가장 많이 사용되는 아이콘을 선별하여 제공한다. 기본으로 제공되는 아이콘은 흰색의 아이콘 모양과 반투명의 검은 원형 배경으로 이루어져있어 어떤 배경색과도 대비를 이룰 수 있다.

기본코드

data-icon 속성값으로 생성할 아이콘 이름을 지정하여 아이콘을 추가한다.



Home

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

아이콘의 위치

기본적으로 버튼에 추가된 아이콘은 버튼내용 왼쪽에 위치한다. data-iconpos 속성을 추가하여 아이콘을 버튼내용 오른쪽, 위쪽, 아래쪽에 위치시킬 수 있다.



왼쪽

오른쪽



위쪽

아래쪽



```
<a href="#" data-role="button" data-icon="gear" data-iconpos="left">왼쪽</a>
<a href="#" data-role="button" data-icon="gear" data-iconpos="right">오른쪽</a>
<a href="#" data-role="button" data-icon="gear" data-iconpos="top">위쪽</a>
<a href="#" data-role="button" data-icon="gear" data-iconpos="bottom">아래쪽</a>
```

data-iconpos 속성값을 "notext"로 지정하여 버튼에 아이콘만 존재하도록 설정할 수 있다.



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

기본 아이콘 20종

data-icon 속성에 각 아이콘의 이름을 넣어 아이콘을 생성한다.



arrow-l



arrow-r



arrow-u




arrow-d



delete







plus

	minus		check
	gear		refresh
	forward		back
	grid		star
	alert		info
	search		home
	phone		mail

사용자 정의 아이콘

전자정부 모바일 표준프레임워크에서 아이콘에 적용하는 기본 CSS 스타일을 변경하여 사용자 정의 아이콘을 사용할 수 있다. 사용자 정의 아이콘(Custom Icon)을 사용하기 위해서는 data-icon 속성값을 고유한 이름으로 지정하여 사용해야 한다. 기존의 아이콘과 통일성을 유지하려면 아이콘의 사이즈를 18X18 픽셀로 지정하고 이미지를 PNG-8, alpha transparency 로 저장한다.

 chat
 magnify
 piggy
 egov

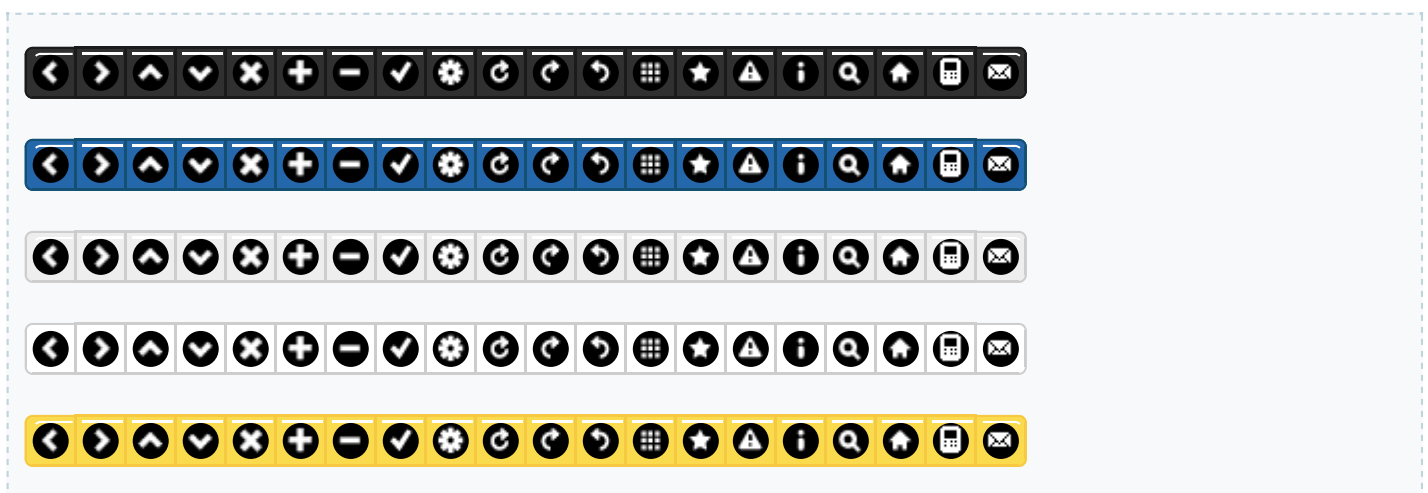
```

.ui-icon-[ideal_icon_name] {
  background-image: url(/images/egovframework/mbl/cmm/[icon_filename]);
  background-repeat: no-repeat;
}

```

아이콘 색상 지정

data-theme 속성에 테마를 적용하여 아이콘의 색상을 변경한다.





```
<a href="#" data-role="button" data-icon="arrow-l" data-iconpos="notext" data-theme="h">arrow-l</a>
```


Selector

Selector의 기본 형식은 OS의 Selector 메뉴 형식에 따라 다르게 나타난다. 또한 팝업 형으로 펼침 방식을 변경 할 수 있으며 <option>의 개수에 따라 자동적으로 형태를 바꿔준다. <label>과 같은 행으로 나타 내고 싶을 때는 <fieldcontain>으로 <label> 과 <selector>를 감싸야 하며 <label>의 for 속성과 Selector의 id 속성 값이 동일 해야한다.

기본코드

<label>과 같은 행으로 나타 내고 싶을 때는 <fieldcontain>으로 <label>과 <select>를 감싸야 하며 <label> for 속성과 <selector>의 id 속성의 값이 동일 해야한다.

선택 하시오:

사과



```
<div data-role="fieldcontain">
<label for="select-choice-1" class="select">선택 하시오:</label>
<select name="select-choice-1" id="select-choice-1">
<option value="apple">사과</option>
<option value="banana">바나나</option>
<option value="grapes">포도</option>
<option value="onion">양파</option>
</select>
</div>
```

모양변경

Selector의 모양을 사각으로 만들고 싶다면 <select> Tag에 data-corners="false"을 추가한다

선택 하시오:

사과



```
<div data-role="fieldcontain">
<label for="select-choice-1" class="select">선택 하시오:</label>
<select name="select-choice-1" id="select-choice-1" data-corners="false">
<option value="apple">사과</option>
<option value="banana">바나나</option>
<option value="grapes">포도</option>
<option value="onion">양파</option>
</select>
</div>
```

Selector의 아이콘을 변경하고 싶다면 <select> Tag에 data-icon="[icon]"을 추가한다

선택 하시오:

사과



```

<div data-role="fieldcontain">
<label for="select-choice-1" class="select">선택 하시오:</label>
<select name="select-choice-1" id="select-choice-1" data-icon="star">
<option value="apple">사과</option>
<option value="banana">바나나</option>
<option value="grapes">포도</option>
<option value="onion">양파</option>
</select>
</div>

```

Selector의 아이콘의 위치를 변경하고 싶다면 <select> Tag에 data-iconpos="[right, left]"을 추가한다

선택 하시오:



사과

```

<div data-role="fieldcontain">
<label for="select-choice-1" class="select">선택 하시오:</label>
<select name="select-choice-1" id="select-choice-1" data-iconpos="left">
<option value="apple">사과</option>
<option value="banana">바나나</option>
<option value="grapes">포도</option>
<option value="onion">양파</option>
</select>
</div>

```

Selector의 크기를 작게 하고 싶다면 <select> Tag에 data-inline="true"을 추가한다

선택 하시오:

사과



```

<div data-role="fieldcontain">
<label for="select-choice-1" class="select">선택 하시오:</label>
<select name="select-choice-1" id="select-choice-1" data-inline="true">
<option value="apple">사과</option>
<option value="banana">바나나</option>
<option value="grapes">포도</option>
<option value="onion">양파</option>
</select>
</div>

```

Selector의 Shadow를 제거하고싶다면 <select> Tag에 data-shadow="false"을 추가한다

선택 하시오:

사과



```

<div data-role="fieldcontain">
<label for="select-choice-1" class="select">선택 하시오:</label>
<select name="select-choice-1" id="select-choice-1" data-shadow="false">
<option value="apple">사과</option>
<option value="banana">바나나</option>
<option value="grapes">포도</option>

```

```

<option value="onion">양파</option>
</select>
</div>

```

형태 변경

Selector option을 그룹화 하여 나타낼 수 있으며 기본코드에 <optgroup label="그룹이름">를 추가하여 그룹화 할 option들을 감싼다.

선택하시오:

사과



```

<div data-role="fieldcontain">
  <label for="select-choice-nc" class="select">선택하시오:</label>
  <select name="select-choice-8" id="select-choice-nc">
    <optgroup label="과일">
      <option value="apple">사과</option>
      <option value="banana">바나나</option>
      <option value="grapes">포도</option>
    </optgroup>
    <optgroup label="야채">
      <option value="onion">양파</option>
      <option value="tomato">토마토</option>
      <option value="carrot" disabled="disabled">당근 (disabled)</option>
      <option value="cabbage">양배추</option>
    </optgroup>
  </select>
</div>

```

Selector를 눌렀을 때 option이 팝업형태와 페이지 형태로 보이도록 한다. 팝업 효과는 기본코드의 <select name> tag에 data-native-menu="false"를 추가하며 option의 길이가 브라우저 넓이를 초과할 시 자동으로 팝업 형태로 출력된다.

선택하시오 :

사과



```

<select name="select-choice-1" id="select-choice-custom" data-native-menu="false">

```

또한 Selector의 기본 선택을 위해서는 첫번째 <option> Tag 에 기본 option을 추가한다.

선택하시오 :



```

<div data-role="fieldcontain">
  <label for="select-choice-5" class="select">선택하시오 : </label>
  <select name="select-choice-5" id="select-choice-5" data-native-menu="false">
    <option></option>
    <option value="apple">사과</option>
    <option value="banana">바나나</option>
    <option value="grapes">포도</option>
    <option value="onion">양파</option>
  </select>
</div>

```

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

option을 비활성화하고 싶다면 <option value>에 disabled="disabled"를 추가한다

선택하시오 :

과일



```
<option value="banana" disabled="disabled">바나나 </option>
```

선택 효과

여러개의 속성을 선택 할수 있으며 Select Tag에 multiple="multiple"을 추가 한다. 이 역시 형태 변경과 동일하게 option의 길이가 브라우저 넓이를 초과 할시 자동으로 팝업 형태로 출력된다

선택 하시오:

과일



```
<div data-role="fieldcontain">
  <label for="select-choice-9" class="select">선택 하시오:</label>
  <select name="select-choice-9" id="select-choice-9" multiple="multiple" data-native-menu="false">
    <option>과일선택</option>
    <option value="apple">사과</option>
    <option value="banana">바나나</option>
    <option value="grapes">포도</option>
    <option value="onion">양파</option>
  </select>
</div>
```

새로고침

Selector는 새로고침 기능이 제공된다. Javascript에 아래 코드를 추가 한 후 이벤트로 연결 해주면 지정한 option으로 돌아간다.

선택하시오 :

과일



refresh

```
var myselect = $("#select#foo");
myselect[0].selectedIndex = 3;
myselect.selectmenu("refresh")
```

Switch

기본 코드

Switch는 이진 형식을 따르며 좌우로 선택 가능 하도록 되어있다. <label>과 같은 행으로 나타 내고 싶을 때는 <fieldcontain>으로 <label>과 <slider>를 감싸야 하며 <label>의 for 속성과 <slider>의 id 속성 값이 동일 해야한다.

선택하시오 :

☐ ☒ Off

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

새로고침

Switch는 새로고침 기능이 제공된다. 자바 스크립트에 아래 코드를 추가 한 후 이벤트로 연결 해주면 지정한 option으로 돌아간다.

선택하시오 :

☐ ☒ Off

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

Collapsible Block

Collapsible Block은 콘텐츠 영역을 접었다 펼 수 있는 컨트롤이다.

기본코드

Collapsible Block을 이용하기 위해서는 블럭을 만들려는 콘텐츠를 감싸고 있는 div Tag에 `>data-role="collapsible"`을 추가 한다. 이때 div Tag안에 처음 정의된 헤더 태그가 collapsible Block의 타이틀로 사용되며 접힘/펼침을 시각적으로 표현하기 위한 `+`, `-` 아이콘이 타이틀 앞에 자동으로 추가 된다. 또한 처음 화면을 로드 할 때 Block이 펼쳐있는 상태를 원할 시에는 감싸고 있는 div Tag에 `data-collapsed="false"`을 추가한다.



default collapsible header

default collapsible header content



open collapsible header

```
<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>
```

색상변경

색상변경 시에는 `data-role="collapsible"`이 선언된 div Tag에 `data-theme=[a~g]`을 선언하여 색상을 변경한다.(총 7가지 색상 지원)



data-theme="b" collapsible header



data-theme="g" collapsible header

data-theme="g" Theme collapsible header content

```
<div data-role="collapsible" data-theme="b">
  <h3>data-theme="b" collapsible header</h3>
  <p>data-theme="b" collapsible header content</p>
</div>
<div data-role="collapsible" data-collapsed="false" data-theme="g">
  <h3>data-theme="g" collapsible header</h3>
  <p>data-theme="g" Theme collapsible header content</p>
</div>
```

형태변경

여러개의 Collapsible Block을 하나의 그룹으로 구성하고 싶다면 이것들을 감싸고 있는 div Tag에 `<div data-role="collapsible-`

set"> 속성을 지정하면 된다.이렇게 그룹으로 구성하면 각 Collapsible Block의 접힘/펼침은 상호 배타적으로 동작한다.

 collapsible 1

 collapsible 2

collapsible 2 content

 collapsible 3

 collapsible 4

```
<div data-role="collapsible-set">
  <div data-role="collapsible">
    <h3>collapsible 1</h3>
    <p>collapsible 1 content</p>
  </div>
  <div data-role="collapsible" data-collapsed="false">
    <h3>collapsible 2</h3>
    <p>collapsible 2 content</p>
  </div>
  <div data-role="collapsible">
    <h3>collapsible 3</h3>
    <p>collapsible 3 content</p>
  </div>
  <div data-role="collapsible">
    <h3>collapsible 4</h3>
    <p>collapsible 4 content</p>
  </div>
</div>
```

연속적인 Collapsible Block을 형성 할때는 content 영역에 하위 Collapsible Block을 선언한다.

 collapsible 1

```
<div data-role="collapsible">
  <h3>collapsible 1</h3>
  <p>collapsible 1 content</p>
  <div data-role="collapsible">
    <h3>collapsible 1-1 content</h3>
    <p>collapsible 1-1 content</p>
  </div>
</div>
```

I 모바일 Page

모바일 Page는 브라우저의 한 화면을 뜻하는 것으로 Header/Footer bar, Content의 UX 컴포넌트 등 이 조합되어 만들어지며 링크를 통한 페이지 전환이 가능하다.

유의사항

각 브라우저마다 history를 관리 하는 방식(pushstate)이 상이 하기 때문에 이를 공통적으로 지원해주기 위하여 history.js를 import하여 사용할 것을 권고한다.

history.js를 import함으로써 기기내 브라우저의 back 및 forward button의 기능이 정상동작한다.

history.js 다운로드

jquery.mobile-1.0b3.min.js 위에 선언

```
<!-- History.js -->
```

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

pushstate 지원 브라우저 : Chrome(5.0), Firefox(4.0), Opera(11.50), safari(5.0)

- 이하버전 브라우저는 history.js를 각 페이지에 import하여 사용

※ 향후 브라우저가 **pushstate**를 지원할 시 해당 스크립트 삭제 가능

기본코드

모바일 Page는 HTML5, 및 CSS3를 기반으로 하고 있으며 문서의 시작은 HTML5의 <!DOCTYPE html> 로 시작하고 <head> 부분에 아래와 같이 선언 한다.

```
<!DOCTYPE html>
<html>
  <head>
    <title>eGovFrame</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no" />
    <link rel="stylesheet" href="/css/egovframework/mb/cmm/jquery.mobile-1.0b3.min.css"/>
    <link rel="stylesheet" href="/css/egovframework/mb/cmm/EgovMobile.css" />
    <script src="/js/egovframework/mb/cmm/jquery-1.6.2.min.js"></script>
    <script src="/js/egovframework/mb/cmm/jquery.mobile-1.0b3.min.js"></script>
    <script src="/js/egovframework/mb/cmm/EgovMobile.js"></script>
  </head>
  <body>
    ...
  </body>
</html>
```

구성 요소

모바일 Page 의 구성요소는 HTML5의 data-* 속성을 사용하여 <body>안에 **header**, content, **footer**를 사용하여 구성한다.

```
<body>
  <!-- 모바일 페이지 start -->
  <div data-role="page">
```



```

<!-- header start -->
<div data-role="header">
  <h1>header Title</h1>
</div>
<!-- header end -->

<!-- content start -->
<div data-role="content">
  <p>전자정부 모바일 표준프레임워크 default Page 화면 입니다.</p>
</div>
<!-- content end -->

<!-- footer start -->
<div data-role="footer">
  <h4>footer title</h4>
</div>
<!-- footer end -->

</div>
<!-- 모바일 페이지 end -->
</body>

```

페이지 Title

페이지 전환 시 Ajax를 기반으로 통신하게 되면 data-role="page" 내의 content만 불러온다. 그래서 기본 웹에서 제공하는 <title>을 변경 할수 없는데 <data-role="page"> Tag 내에 data-title="[title]" 속성을 지정하면 활성화된 page의 title을 지정할 수 있다.

```

<div data-role="page" id="foo" data-title="Page Foo">
<div

```

페이지 로딩

모바일 전자정부 표준 프레임워크는 페이지 내부 링크 보다는 도메인 내부 링크를 권장한다.

Prefetch 페이지

도메인 내부링크를 사용하게 되면 내부 링크 보다는 빠른 화면 전환 및 데이터 로딩을 할수 없기때문에 data-prefetch 을 선언한다. 연결하는 도메인 내부 링크에 data-prefetch를 선언하면 기본 페이지를 로딩, data-prefetch가 선언된 페이지를 읽기 위해 pagecreate 이벤트가 발생 한다.그 후 백그라운드로 대상 페이지를 로드한다. 연결 전 페이지를 미리 로드해 놓기 때문에 빠른 화면 전환이 가능하다.

```

<a href="/guide/components/pageStructure/page.do" data-prefetch> ... </a>

```

Caching 페이지

모바일 전자정부 페이지는 기본적으로 페이지 전환이 되면 이전 페이지의 DOM은 삭제한다. 그러나 이전 페이지의 DOM을 삭제하고 싶지 않다면 domCache에 저장되도록 한다. 하지만 저장되는 DOM의 크기가 클 경우 화면 전환 문제 및 메모리 이슈가 있을 수 있기 때문에 필요한 경우에만 사용하도록 권장한다.

```

< mobileinit에 선언 할 경우>
$.mobile.page.prototype.options.domCache = true;

```

<페이지에 선언할 경우>

```
<div data-role="page" id="cacheMe" data-dom-cache="true">
```

Ajax를 통한 Hash URL 제어

Ajax 통신 방식을 할 경우 hash url이 나타나는 것을 설정을 통하여 제어 할 수 있다. hash url이 필요할 때는 \$.mobile.pushStateEnabled=true hash url이 필요 없다면 \$.mobile.pushStateEnabled=false을 선언한다.

*- default는 \$.mobile.pushStateEnabled=true

< mobileinit에 선언>

```
$.mobile.pushStateEnabled=true;
```

페이지 이동

모바일 페이지 이동은 기본적으로 Ajax 를 이용하여 처리 된다. 이는 모바일에 최적화된 화면 전환 효과를 주기 위함으로 옵션 설정을 통해 변경 가능하다.

페이지 내부 이동

한 HTML 안에 여러 page 가 선언 되어 있을 경우 사용 할 수 있는 방법으로 모바일 page 구성의 기본 방식이다. 페이지 내부 이동은 Ajax 통신을 사용하며 page 로 선언된 div 태그의 id 값을 링크(#pageId)에 적용하여 사용 가능하다.

* 한 HTML 내에 여러 page가 선언 되어 있을 경우 제일 상단의 page를 첫 화면으로 인식 한다.

```
<!-- main page -->
```

```
<div id="main" data-role="page">
```

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

```
<p>Main Page 입니다.</p></div>
```

```
<!-- view page -->
```

```
<div id="view" data-role="page">
```

```
  <p>상세보기 페이지</p>
```

```
</div>
```

페이지 외부 이동

페이지 외부 이동은 Ajax 통신을 이용하며 Ajax 로 호출한 html 의 data-role="page" 영역만 읽어 들어서 호출한 html 페이지의 DOM 요소에 추가 해 준다.(페이지 내부 이동과 유사 한 구조로 DOM 관리)

* Ajax 로 호출된 HTML의 page 영역만 가져 오기 때문에 호출된 페이지에서 사용하는 Javascript, CSS 등은 호출을 한 HTML내에 존재 해야 한다.

* Ajax 통신을 사용하고 싶지 않은 경우 [Link](#)를 참조하여 변경 가능하다.

main.html

```
<div id="main" data-role="page">
```

```
  <a href="create.html" data-role="button">글쓰기</a>
```

```
<p>Main Page 입니다.</p></div>
```

...

create.html

```
<div id="create" data-role="page">
```

```
  <p>글쓰기 페이지</p>
```

```
</div>
```

...

Header bar

Header bar는 일반적으로 페이지간 이동을 위하여 브라우저 상단에 위치하며 좌우에 버튼 및 헤더제목틀 넣을 수 있다.

기본코드

Header bar는 기본적으로 검정색으로 나타나며 Header bar의 높이는 Title 크기에 따라 달라진다.

Header title

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

색상 변경

기본 코드 <data-role="header"> Tag에 data-theme='[적용테마: a~g]'을 추가한다.

data-theme="a"

data-theme="b"

data-theme="c"

data-theme="d"

data-theme="e"

data-theme="f"

data-theme="g"

```
<div data-role="header" data-theme="e">
  <h1>data-theme="e"</h1>
</div>
```

버튼 추가

Header bar 에 Anchor 링크를 추가하면 자동으로 버튼이 생성된다. 왼쪽부터 버튼이 나타나며 2개 이상의 버튼은 생성할수 없다. 기본형 <data-role="header"> data-position="inline"을 넣어 버튼이 잘 정렬 되도록 한다. 버튼에 모양은 ICon 를 참조 한다.

Cancel

Add button in header bar

Save

```
<div data-role="header" data-position="inline">
  <a href="#" data-icon="delete">Cancel</a>
  <h1>Add button in header bar</h1>
  <a href="#" data-icon="check" data-theme="b">Save</a>
```

</div>

버튼 위치

Header bar에서 버튼의 위치를 지정 할 때는 버튼에 class="ui-btn-[right, left]"를 추가 한다.

class="ui-btn-right"



```
<div data-role="header" data-position="inline" >
  <h1>class="ui-btn-right"</h1>
  <a href="index.html" data-icon="gear" class="ui-btn-right">Options</a>
</div>
```

뒤로 가기

Header bar에 뒤로 가기 버튼을 생성하려면 data-rel="back"을 추가 한다.



Back

```
<a href="#" data-icon="arrow-l" data-rel="back">back</a>
```

data-rel="back"은 브라우저 지원 요소에 따라 다르게 동작 할 수도 있어 이전 화면의 링크를 걸어 준 후 data-direction="reverse"를 선언 하면 뒤로 가는 화면 전환 효과와 함께 링크가 정확하게 동작 된다.



Back

```
<a href="/guide/guide.do" data-icon="arrow-l" data-direction="reverse" >back</a>
```

위치 선정

고정 위치

Header bar의 위치를 브라우저 맨 위/ 끝으로 고정된다. 스크롤 이벤트가 일어날 경우 Header/footer bar는 항상 브라우저의 맨 위와/끝에 나타난다. Header bar의 <div data-role="Header">에 data-position="fixed"를 추가 한다 [fixed position 예시 보기](#)

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

전체 화면 모드

Header/footer bar의 위치를 브라우저 맨 위/ 끝으로 고정하거나 없앨 수 있다. 이벤트가 일어날 경우 Header/footer bar가 나타나거나 없어진다. Header/footer bar의 fixed position과 같은 형태에서 <div data-role="page"> Tag에 data-fullscreen="true"을 추가 한다. [fullscreen position 예시 보기](#)

```
<div data-role="page" data-fullscreen="true">
  <div data-role="header" data-position="fixed">
    <h1>header title</h1>
  </div>
  ...
```

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

Footer bar

Footer bar는 브라우저 하단에 위치하며 버튼, 탭, 링크 등 컴포넌트 요소 등을 넣을 수 있다.

기본코드

Footer bar는 기본적으로 검정색으로 나타나며 Footer bar의 높이는 Title 크기에 따라 달라진다.

Footer title

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

색상 변경

기본 코드 `<data-role="footer">` Tag에 `data-theme='[적용테마: a~g]'` 을 추가한다. 색상 유형은 [Header bar의 색상변경](#)을 참고한다.

data-theme="b"

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

버튼 추가

Footer bar에 Anchor 링크를 추가하면 자동으로 버튼이 생성된다. 버튼은 여러개 생성 할수 있으며 버튼은 자동으로 오른쪽 정렬 된다. 버튼에 모양은 [UX Component ICon](#) 부분을 참조 한다.

Remove Add Up Down

```
<div data-role="footer">
  <a href="index.html" data-role="button" data-icon="delete">Remove</a>
  <a href="index.html" data-role="button" data-icon="plus">Add</a>
  <a href="index.html" data-role="button" data-icon="arrow-u">Up</a>
  <a href="index.html" data-role="button" data-icon="arrow-d">Down</a>
</div>
```

버튼 정렬

Footer bar에서 버튼을 그룹화 할때는 Footer에 `<div data-role="controlgroup" data-type="horizontal">` 로 버튼을 감싸준다

Remove Add Up Down

```
<div data-role="footer">
  <div data-role="controlgroup" data-type="horizontal">
    <a href="index.html" data-role="button" data-icon="delete">Remove</a>
    <a href="index.html" data-role="button" data-icon="plus">Add</a>
  </div>
</div>
```

```

<a href="index.html" data-role="button" data-icon="arrow-u">Up</a>
<a href="index.html" data-role="button" data-icon="arrow-d">Down</a>
</div>
</div>

```

컴포넌트 추가

Footer bar에는 UX 컴포넌트를 추가 하여 화면 전환 및 조정을 용이하게 할 수 있다.

Tab 컴포넌트

Footer bar의 위치에 [Tab 컴포넌트](#)를 넣어 화면간 이동을 용이하게 할 수 있다. [Footer Tab 예시 보기](#)

```


<div data-role="footer" data-theme="z" data-position="fixed">
  <div data-role="navbar">
    <ul>
      <li><a href="exHome.jsp">Home</a></li>
      <li><a href="header.jsp">Header bar </a></li>
      <li><a href="footer.jsp">Footer bar</a></li>
    </ul>
  </div>
</div>

```

기타 컴포넌트

Footer bar에 다른 컴포넌트를 넣을 수 있다.

선택하시오:



```

<div data-role="footer" class="ui-bar" data-position="inline">
  <label for="select-choice-1">Shipping:</label>
  <select name="select-choice-1" id="select-choice-1" data-theme="a">
    <option value="apple">사과</option>
    <option value="banana">바나나</option>
    <option value="grapes">포도</option>
    <option value="onion">양파</option>
  </select>
</div>

```

위치 선정

고정 위치

footer bar의 위치를 브라우저 맨 위/ 끝으로 고정된다. 스크롤 이벤트가 일어날 경우 Footer bar는 항상 브라우저의 맨 위와/끝에 나타난다. Header bar의 <div data-role="footer">에 data-position="fixed"를 추가 한다 [fixed position 예시 보기](#)

```

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

```

전체 화면 모드

Header/footer bar의 위치를 브라우저 맨 위/ 끝으로 고정하거나 없앨 수 있다. 이벤트가 일어날 경우 Header/footer bar가 나타나거나 없어진다. 자세한 내용은 [header bar](#)의 전체 화면 모드를 참고 한다.

Theme

Theme 기능을 이용하여 하위 레벨의 색상을 변경할 수 있다. 최상위 테마 변경 Tag는 `<data-role="page">` 이며 변경시 page 하위의 Header, Footer, Panel, UX Component의 색상이 변한다.

기본코드

테마를 적용할 최상위 div Tag에 `data-theme="[a~g]"`를 추가한다.

Header Title

Page 테마 적용시
Header, Panel, Footer 의 색상이 변경된다.

Footer Title

```
<div data-role="header" data-theme="e">
  <h1>Header Title</h1>
</div>
<div data-role="content">
  <p>Page 테마 적용시
  Header, Panel, Footer 의 색상이 변경된다.</p>
</div>
<div data-role="footer" data-theme="e">
  <h1>Footer Title</h1>
</div>
```

테마 예제

테마를 적용 할 경우 하위 컴포넌트들의 색상이 일괄 변형된다. 만약 해당 상위테마를 사용하고 싶지 않은 경우 적용하고 싶지 않은 요소의 최상위 div Tag에 다른 테마를 정의한다.

테마 선택 :

`data-theme="a"`



a 테마 적용 (`data-theme="a"`)

■ Header

New

Bar theme "a"

Save

```
<div data-role="header" data-position="inline" data-theme="a">
  <a href="#" data-icon="plus">New</a>
  <h1>Bar theme "a"</h1>
  <a href="#" data-icon="arrow-r" data-iconpos="right">Save</a>
</div>
```

■ Footer



```
<div data-role="footer" data-theme="a">
  <a href="#" data-role="button" data-icon="arrow-l" data-iconpos="right">left</a>
  <a href="#" data-role="button" data-icon="arrow-r" data-iconpos="right">right</a>
  <a href="#" data-role="button" data-icon="arrow-u" data-iconpos="right">up</a>
  <a href="#" data-role="button" data-icon="arrow-d" data-iconpos="right">down</a>
</div>
```

■ Button



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

■ Panel



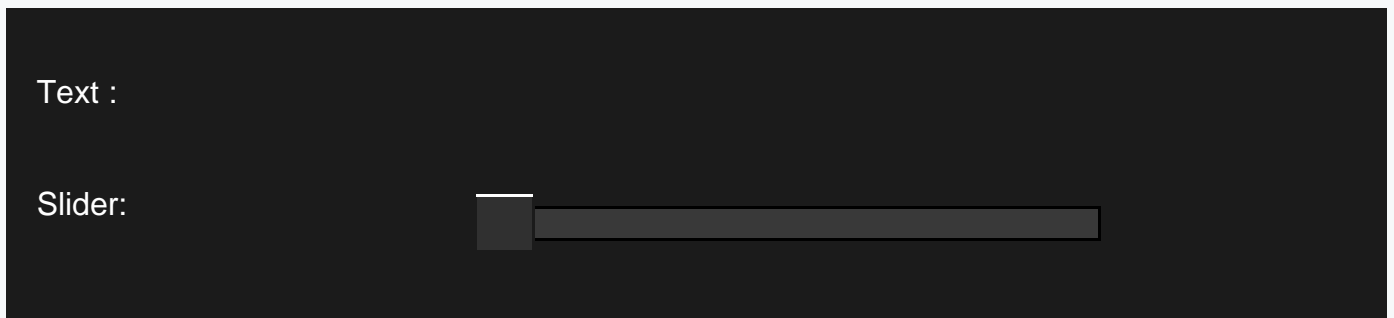
```
<div class="ui-bar-a">
  <h4>Panel a</h4>
</div>
```

■ Tabs



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

■ Form



```
<form>
  <div data-role="content" data-theme="a">
    <div data-role="fieldcontain">
      <label for="name">Text :</label>
      <input type="text" id="name"/>
    </div>
  </div>
```

```

</div>
<div data-role="fieldcontain">
  <label for="slider">Slider:</label>
  <input type="range" id="slider" value="0" min="0" max="100" />
</div>
</div>
</form>

```

■ Menu

Menu

전자정부 모바일 표준프레임워크 개요

```

<ul data-role="listview" data-inset="true" data-theme="a" data-dividertheme="a">
  <li data-role="list-divider">Menu</li>
  <li><a href="#menu1" data-rel="dialog" data-transition="slide">전자정부 모바일 표준프레임워크 개요</a></li>
</ul>

```

■ Dialog

ActionSheet Dialog data-theme="a"

Overlay Dialog data-theme="a"

Alert Dialog data-theme="a"

Confirm Dialog data-theme="a"

Show prompt Dialog data-theme="a"

■ Grid

ui-block-a

ui-block-b

```

<div class="ui-grid-a">
  <div class="ui-block-a"><div class="ui-body-a" align="center"><h3>ui-block-a</h3></div></div>
  <div class="ui-block-b"><div class="ui-body-a" align="center"><h3>ui-block-b</h3></div></div>
</div>

```

■ List

Nested List

Numbered List

Split Button

List Divider

Count Bubble

Thumbnail

List Icon

Content Formatting

Search Filter Bar

```
<ul data-role="listview" data-inset="true" data-theme="a">
  <li>Nested List</li>
  <li>Numbered List</li>
  <li>Split Button</li>
  <li>List Divider</li>
  <li>Count Bubble</li>
  <li>Thumbnail</li>
  <li>List Icon</li>
  <li>Content Formatting</li>
  <li>Search Filter Bar</li>
</ul>
```

■ Date / Time Picker

안드로이드 형태

달력 형태

```
<div data-role="fieldcontain">
  <label for="andate">안드로이드 형태</label>
  <input name="mydate" id="andate" type="date" data-role="datebox" data-options='{ "page": "a", "button": "a" }'>
</div>
<div data-role="fieldcontain">
  <label for="caldate">달력 형태</label>
  <input name="mydate" id="caldate" type="date" data-role="datebox" data-options='{ "mode": "calbox", "button": "a" }'>
</div>
```

■ Check Box

Check Box :

check 1	check 2	check 3
---------	---------	---------

```
<div data-role="fieldcontain">
  <fieldset data-role="controlgroup" data-type="horizontal">
```

```

<legend data-theme="a">Check Box :</legend>
<input type="checkbox" name="checkbox-6" id="checkbox-6" class="custom" />
<label for="checkbox-6" data-theme="a">check 1</label>
<input type="checkbox" name="checkbox-7" id="checkbox-7" class="custom" />
<label for="checkbox-7" data-theme="a">check 2</label>
<input type="checkbox" name="checkbox-8" id="checkbox-8" class="custom" />
<label for="checkbox-8" data-theme="a">check 3</label>
</fieldset>
</div>

```

Radio

Radio Button:

```

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

```

Icon



```

<div data-role="controlgroup" data-type="horizontal">
  <a href="#" data-role="button" data-icon="arrow-l" data-iconpos="notext" data-theme="a">arrow-l</a>
  <a href="#" data-role="button" data-icon="arrow-r" data-iconpos="notext" data-theme="a">arrow-r</a>
  <a href="#" data-role="button" data-icon="arrow-u" data-iconpos="notext" data-theme="a">arrow-u</a>
  <a href="#" data-role="button" data-icon="arrow-d" data-iconpos="notext" data-theme="a">arrow-d</a>
  <a href="#" data-role="button" data-icon="delete" data-iconpos="notext" data-theme="a">delete</a>
  <a href="#" data-role="button" data-icon="plus" data-iconpos="notext" data-theme="a">plus</a>
  <a href="#" data-role="button" data-icon="minus" data-iconpos="notext" data-theme="a">minus</a>
  <a href="#" data-role="button" data-icon="check" data-iconpos="notext" data-theme="a">check</a>
  <a href="#" data-role="button" data-icon="gear" data-iconpos="notext" data-theme="a">gear</a>
  <a href="#" data-role="button" data-icon="refresh" data-iconpos="notext" data-theme="a">refresh</a>
  <a href="#" data-role="button" data-icon="forward" data-iconpos="notext" data-theme="a">forward</a>
  <a href="#" data-role="button" data-icon="back" data-iconpos="notext" data-theme="a">back</a>
  <a href="#" data-role="button" data-icon="grid" data-iconpos="notext" data-theme="a">grid</a>
  <a href="#" data-role="button" data-icon="star" data-iconpos="notext" data-theme="a">star</a>
  <a href="#" data-role="button" data-icon="alert" data-iconpos="notext" data-theme="a">alert</a>
  <a href="#" data-role="button" data-icon="info" data-iconpos="notext" data-theme="a">info</a>
  <a href="#" data-role="button" data-icon="search" data-iconpos="notext" data-theme="a">search</a>
  <a href="#" data-role="button" data-icon="home" data-iconpos="notext" data-theme="a">home</a>
  <a href="#" data-role="button" data-icon="phone" data-iconpos="notext" data-theme="a">phone</a>
  <a href="#" data-role="button" data-icon="mail" data-iconpos="notext" data-theme="a">mail</a>
</div>

```

Selector / Switch

Select Menu:

choice 1



```
<div data-role="fieldcontain">
  <label for="select-choice-1" class="select">Select Menu:</label>
  <select id="select-choice-1" data-theme="a">
    <option value="choice1">choice 1</option>
    <option value="choice2">choice 2</option>
    <option value="choice3">choice 3</option>
    <option value="choice4">choice 4</option>
  </select>
</div>
```

Switch :

Off

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

 collapsible 1

 collapsible 2

collapsible 2 content

```
<div data-role="collapsible-set">
  <div data-role="collapsible" data-theme="a">
    <h3>collapsible 1</h3>
    <p>collapsible 1 content</p>
  </div>
  <div data-role="collapsible" data-collapsed="false" data-theme="a">
    <h3>collapsible 2</h3>
    <p>collapsible 2 content</p>
  </div>
</div>
```