



31 모바일 웹

모던 웹을 위한 Javascript jQuery 입문

❖ 한 번의 개발로 모든 운영체제 지원하기

■ HTML을 사용하면 가능

- 웹 페이지면서도 일반 상용 애플리케이션보다 더 사용하기 편리하기도.
- 모든 스마트 폰에서 동작하는 웹 페이지

그림 31-1 모바일 전용 웹스폰과 미투데이 웹 사이트



❖ Photo Gap

- 웹 애플리케이션을 스마트폰 애플리케이션으로 변환해주는 프로그램

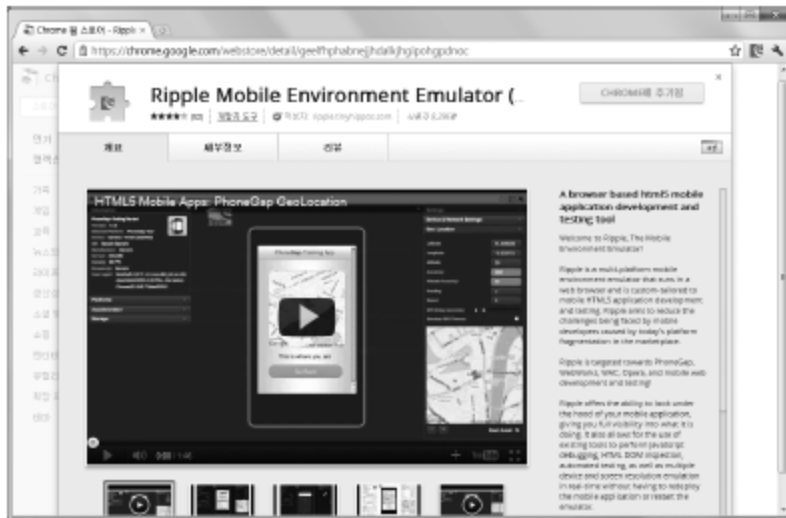
그림 31-2 Phone Gap



❖ Ripple Mobile

- 데스크톱 내에서도 에뮬레이터를 사용해 스마트 폰과 비슷하게 확인
- 크롬 웹 브라우저로 크롬 웹 스토어 (<https://chrome.google.com/webstore>) 에 접속한 뒤 검색
- 크롬 웹 스토어에 있는 제품이므로 크롬에서만 사용 가능

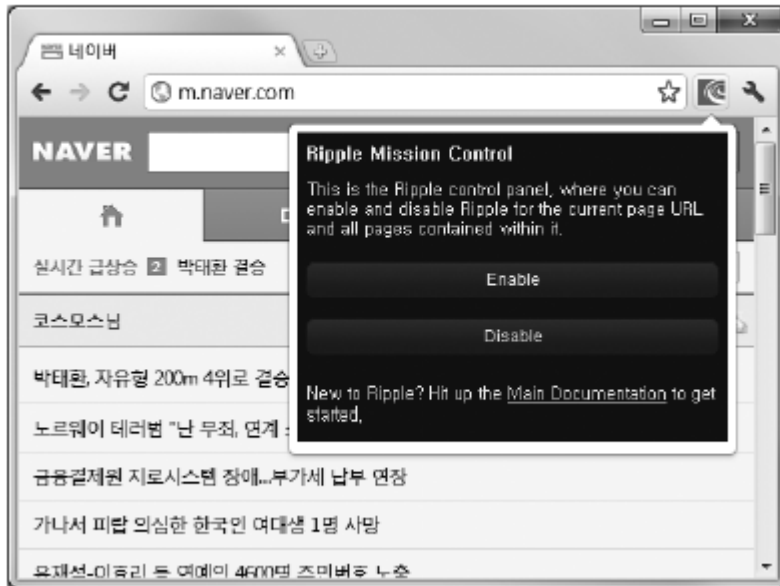
그림 31-3 Ripple Mobile Environment Emulator



❖ 에뮬레이터 설치 후 사용

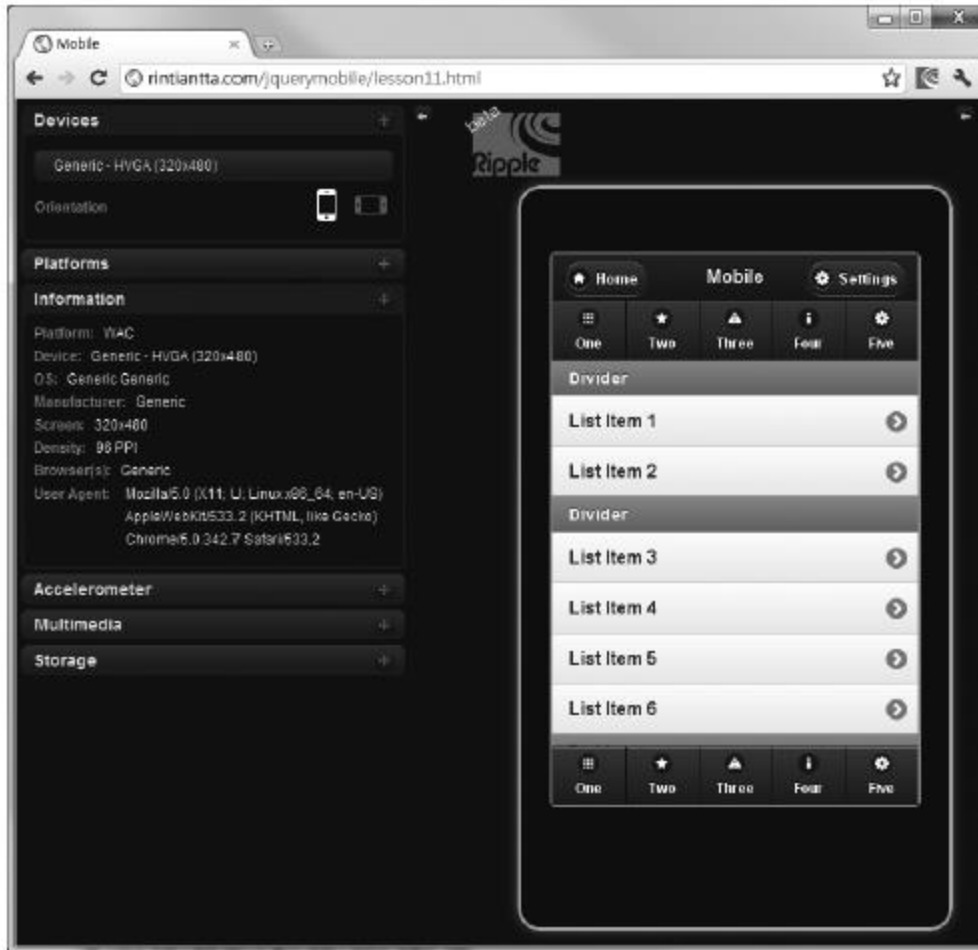
- 크롬의 오른쪽 위 새로 생긴 아이콘 클릭
- 그림 31-4처럼 이 아이콘을 누르고 Enable 선택
- 웹 페이지를 모바일처럼 볼 수 있음

그림 31-4 사용 방법



❖ Ripple 에뮬레이터 사용화면 예시

그림 31-5 예시



❖ 스마트폰 확인 방법

- 스마트폰으로 <http://naver.com> 접속
 - 자동으로 <http://m.naver.com> 이동
 - 사용자가 스마트폰을 사용하는지 확인해 스마트폰이면 <http://m.naver.com>으로 이동
- navigator 객체의 userAgent 속성 사용
 - userAgent 속성 - 현재 웹 페이지를 실행하는 웹 브라우저의 정보 표시



31.2 모바일 장치 구분

❖ 예제 코드 31-1]

- userAgent가 어떠한 값을 가지고 있는지 확인

코드 31-1 userAgent 속성

```
<script>
    var userAgent = navigator.userAgent;
    alert(userAgent);
</script>
```

그림 31-6 아이폰과 윈도우폰 7의 실행 결과



❖ usetAgent 속성

- 스마트폰 브라우저의 usetAgent 속성
 - 브라우저 구분할 수 있는 고유한 문자열 존재
 - iPhone
 - Opera Mini
 - iPod
 - Opera Mobi
 - iPad
 - Nokia
 - Android
 - WebOS
 - Windows CE
 - BlackBerry
 - IEMobile
 - SonyErricsson



❖ 예제 코드 31-2]

- 앞 페이지의 키워드 사용해 처럼 스마트 폰 종류 구분하는 코드 작성

코드 31-2 스마트폰 구분

```
<script>
    // 변수를 선언합니다.
    var smartPhones = [
        'iphone', 'ipod',
        'windows ce',
        'android', 'blackberry',
        'nokia', 'webos',
        'opera mini', 'sonyerricsson',
        'opera mobi', 'iemobile'
    ];

    for (var i in smartPhones) {
        // 스마트폰 확인
        if (navigator.userAgent.toLowerCase().match(new RegExp(smartPhones[i]))) {
            alert('This is Smart Phone..!');
            document.location = 'http://m.hanb.com';
        }
    }
</script>
```



❖ 모바일 장치 구분

- 데스크톱에서 실행하면 아무 반응 없음
- 문자열 Windows Mobile
 - Windows Mobile 6.5 이전의 기기
- 문자열 Windows Phone
 - Windows Phone 7 이상 버전



❖ viewport meta 태그

- meta 태그 - 웹 브라우저에 특별한 정보를 제공하는 HTML 태그
- 모바일 웹 페이지에서 화면과 관련된 정보 제공 위해 사용
- Ex) 네이버 모바일 페이지
 - <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0, user-scalable=no, target-densitydpi=medium-dpi"/>
- viewport meta 태그의 content
 - 일부 모바일 브라우저에서는 target-densitydpi 속성 지원하지 않음

표 31-1 viewport meta 태그의 속성

속성 이름	예	설명
width	width=240	화면의 너비
	width=device-width	
height	height=800	화면의 높이
	height=device-height	
initial-scale	initial-scale=2.0	초기 확대 비율
user-scalable	user-scalable=no	확대 및 축소의 가능 여부
minimum-scale	minimum-scale=1.0	최소 축소 비율
maximum-scale	maximum-scale=2.0	최대 축소 비율
target-densitydpi	target-densitydpi=medium-dpi	DPI 지정



31.3 viewport meta 태그

❖ 예제에 사용할 viewport meta 태그

- 화면의 크기 변하지 않게 설정
- maximum-scale 속성, minimum-scale 속성 입력할 필요 없음

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

❖ viewport meta 태그 설명 참조 링크

- 안드로이드
 - <http://developer.android.com/guide/webapps/targeting.html>
- 아이폰
 - <http://developer.apple.com/library/safari/#documentation/appleapplications/reference/safarihtmlref/Articles/MetaTags.html>



❖ 모바일 브라우저

- 데스크톱이나 노트북의 브라우저에 없는 이벤트 두 가지
 - 방향 전환 이벤트와 터치 이벤트

❖ orientationchange 이벤트

- 방향 전환 이벤트의 정식 명칭
- window 객체에 있는 이벤트
 - bind() 메서드 사용

코드 31-3 orientationchange 이벤트

```
<script>
$(document).ready(function () {
    $(window).bind('orientationchange', function () {
        alert(window.orientation);
    });
});
</script>
```

그림 31-7 실행 결과



31.4 화면 방향 전환 이벤트

❖ orientation 속성

- 화면 회전각 의미
- 표 31-2의 네 가지 값으로 화면의 상태 구분

표 31-2 orientation 속성값에 따른 장치의 상태

각도	의미
0	수직
90	왼쪽으로 수평
-90	오른쪽으로 수평
180	역 방향으로 수직



31.4 화면 방향 전환 이벤트

❖ 예제 코드 31-4]

- 애플 웹 지원 프로그램에 있는 예제를 jQuery로 변형한 것
- switch 조건문을 사용해 화면의 각도 따라 다른 내용 출력

코드 31-4 window 객체의 orientation 속성

```
<script>
// 윈도우가 회전될 때
$(window).bind('orientationchange', function () {
    var output = '<h1>Orientation</h1>';

    // 회전 각도에 따라 조건
    switch (window.orientation) {
        case 0:
            output += '<h2>Portrait</h2>';
            break;
        case -90:
            output += '<h2>Landscape</h2>';
```



31.4 화면 방향 전환 이벤트

❖ 예제 코드 31-4] (2)

```
        output += '<h2>right, screen turned clockwise</h2>';  
        break;  
    case 90:  
        output += '<h2>Landscape</h2>';  
        output += '<h2>left, screen turned counterclockwise</h2>';  
        break;  
    case 180:  
        output += '<h2>Portrait</h2>';  
        output += '<h2>upside-down portrait</h2>';  
        break;  
    }  
    // 출력  
    $('body').html(output);  
});  
</script>
```

그림 31-8 방향 전환 실행 결과



❖ 터치 이벤트

- 말그대로 손으로 만지면 발생하는 이벤트
- W3C는 표 31-3의 터치 이벤트를 표준으로 정의
 - touchcancel 이벤트는 현재 touchend 이벤트와 차이가 없음
 - 터치 이벤트와 마우스 이벤트를 동시에 지원하는 경우
 - 터치 이벤트만 지원하는 경우
 - 마우스 이벤트만 지원하는 경우 (윈도우폰 7의 현재 버전 망고)
 - 모바일 애플리케이션을 만들 때는 두 가지 이벤트를 모두 지원하도록
 - 터치 이벤트와 마우스 이벤트의 차이 - 인식 점의 개수

표 31-3 터치 이벤트

이벤트 이름	설명
touchstart	터치가 시작할 때 발생합니다.
touchend	터치가 종료할 때 발생합니다.
touchmove	터치한 상태로 이동할 때 발생합니다.
touchenter	터치가 요소의 경계 외부에서 내부로 이동할 때 발생합니다.
touchleave	터치가 요소의 경계 내부에서 외부로 이동할 때 발생합니다.



❖ 터치 이벤트 테스트

- 간단한 이벤트 연결 방식 사용 불가
 - orientationchange 이벤트와 마찬가지로 bind() 메서드 사용

코드 31-5 HTML 페이지 구성

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-1.6.2.min.js"></script>
  <script>

  </script>
</head>
<body>

</body>
</html>
```



❖ 예제 코드 31-6]

- bind() 메서드를 사용해 touchstart, touch move, touchend 연결

코드 31-6 터치 이벤트 연결

```
<script>
$(document).ready(function () {
    $(document).bind('touchstart', function (e) {
        $('body').append('<h1>touch Start</h1>');
    });

    $(document).bind('touchmove', function (e) {
        $('body').append('<h1>touch Move</h1>');
    });

    $(document).bind('touchend', function (e) {
        $('body').append('<h1>touch End</h1>');
    });
});
</script>
```



❖ 터치 이벤트

- jQuery 현재 1.7 버전에서 지원하지 않음 – 객체 이용 불가
 - jQuery 이벤트 객체
 - **표준 이벤트 객체를 가공해 모든 브라우저에서 사용 가능**
 - **원래의 표준 이벤트 객체 사용**
- 예제 코드 31-7
 - jQuery 이벤트 객체의 originalEvent 속성을 사용 – 표준 이벤트 객체 사용

코드 31-7 originalEvent 속성

```
<script>
$(document).ready(function () {
    $(document).bind('touchstart', function (e) {
        var event = e.originalEvent;
    });

    $(document).bind('touchmove', function (e) {
        var event = e.originalEvent;
    });

    $(document).bind('touchend', function (e) {
        var event = e.originalEvent;
    });
});
</script>
```



❖ 터치 이벤트

- touches 속성은 배열
 - 아이폰이나 안드로이드 3.0 이상에서만 여러 점의 터치 이벤트 인식

❖ 예제 코드 31-8

- 이벤트 객체의 touches 속성 0번째에 위치한 인식점 사용

코드 31-8 touches 속성

```
<script>
$(document).ready(function () {
    $(document).bind('touchstart', function (e) {
        var event = e.originalEvent;
        $('#output').html('<h1>' + event.touches[0].screenX + '</h1>');
        e.preventDefault();
    });

    $(document).bind('touchmove', function (e) {
        var event = e.originalEvent;
        $('#output').html('<h1>' + event.touches[0].screenX + '</h1>');
        e.preventDefault();
    });

    $(document).bind('touchend', function (e) {
        var event = e.originalEvent;
        $('#output').html('<h1>' + event.touches[0].screenX + '</h1>');
        e.preventDefault();
    });
});
</script>
```



❖ 터치 이벤트의 기본 이벤트 리스트

표 31-4 터치 객체의 속성

속성 이름	설명
identifier	인식 점을 구분하기 위한 인식 점 번호
screenX	사용자 화면을 기준으로 한 X 좌표 위치
screenY	사용자 화면을 기준으로 한 Y 좌표 위치
clientX	브라우저의 화면을 기준으로 한 X 좌표 위치
clientY	브라우저의 화면을 기준으로 한 Y 좌표 위치
pageX	브라우저의 화면을 기준으로 한 X 좌표 위치
pageY	브라우저의 화면을 기준으로 한 Y 좌표 위치



❖ jQuery Mobile (jQuery 플러그인)

- 모바일 웹 페이지를 생성시 사용
- 범용적인 기능을 한꺼번에 제공 - 프레임워크로 취급
- jQuery Mobile의 기본 사용 방법
 - jQuery UI와 매우 비슷
 - 공식 홈페이지 <http://jquerymobile.com> 에서 다운

그림 31-9 jQuery Mobile 사이트(<http://jquerymobile.com/>)



❖ jQuery Mobile 사이트의 Download 탭

- 그림 31-10처럼 간단한 사용 방법
 - CDN 호스트를 사용하는 방식
 - 직접 다운받아 사용
- CDN 호스트를 이용하려면?
 - 그림 31-10에 위치한 Copy-and- Paste Snippet 부분을 복사해 사용
- jQuery Mobile와 Phone Gap 등의 프레임워크 함께 사용
 - 하이브리드 애플리케이션 제작할 때에는 오프라인 상태에서 jQuery Mobile 을 사용할 수 있어야 하므로 다운받아 사용

그림 31-10 jQuery Mobile의 CDN 호스트

CDN-Hosted JavaScript:

- Uncompressed: jquery.mobile-1.0rc2.js (useful for debugging)
- Minified and Gzipped: jquery.mobile-1.0rc2.min.js (24KB, ready to deploy)

CDN-Hosted CSS:

- Uncompressed: jquery.mobile-1.0rc2.css (useful for debugging)
- Minified and Gzipped: jquery.mobile-1.0rc2.min.css (7KB, ready to deploy)
- Uncompressed structure without a theme: jquery.mobile-1.0rc2.css (useful for theme development)
- Minified and Gzipped structure without a theme: jquery.mobile-1.0rc2.min.css (6KB, ready to deploy)

Copy-and-Paste Snippet for CDN-hosted files (recommended):

```
<link rel="stylesheet" href="http://code.jquery.com/mobile/1.0rc2/jquery.mobile-1.0rc2.min.css" />
<script src="http://code.jquery.com/jquery-1.6.4.min.js"></script>
<script src="http://code.jquery.com/mobile/1.0rc2/jquery.mobile-1.0rc2.min.js"></script>
```



❖ jQuery Mobile 사용 위한 태그 구성

코드 31-9 head 태그 구성

```
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0,
    user-scalable=no" />
  <link rel="stylesheet" href="http://code.jquery.com/mobile/1.0/jquery.
    mobile-1.0.min.css" />
  <style>

  </style>
  <script src="http://code.jquery.com/jquery-1.7.js">/script>
  <script>

  </script>
  <script src="http://code.jquery.com/mobile/1.0/jquery.mobile-1.0.min.js">/script>
</head>
```



❖ 페이지

- jQuery Mobile의 가장 기본적인 구성 요소
- 페이지를 만들려면?
 - div 태그를 만들고 div의 data-role 속성에 page 입력
- 페이지 안은 헤더와 콘텐츠, 푸터로 레이아웃 구성
 - 같은 방법으로 div 태그의 data-role 속성에 각기 header, content, footer

코드 31-10 페이지

```
<body>
  <div data-role="page">

    </div>
</body>
```



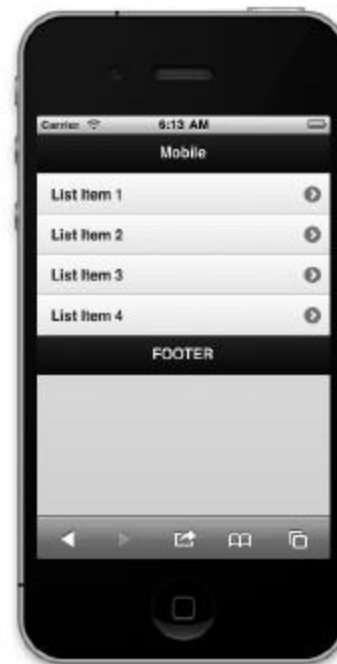
❖ 예제 코드 31-11]

- jQuery Mobile의 화면 구성
 - 내용을 입력하고 실행

코드 31-11 페이지 구성

```
<body>
  <div data-role="page">
    <div data-role="header">
      <h1>Mobile</h1>
    </div>
    <div data-role="content">
      <ul data-role="listview">
        <li><a href="#">List Item 1</a></li>
        <li><a href="#">List Item 2</a></li>
        <li><a href="#">List Item 3</a></li>
        <li><a href="#">List Item 4</a></li>
      </ul>
    </div>
    <div data-role="footer">
      <h1>FOOTER</h1>
    </div>
  </div>
</body>
```

그림 31-11 기본적인 페이지



❖ 내부 페이지 이동

- 여러 개의 페이지를 생성하고 a 태그의 href 속성에 id 입력

코드 31-12 내부 페이지 이동

```
<body>
  <!-- 첫 번째 페이지 -->
  <div data-role="page" id="first_page">
    <div data-role="header">
      <h1>Mobile</h1>
    </div>
    <div data-role="content">
      <ul data-role="listview">
        <li><a href="#second_page">List Item 1</a></li>
        <li><a href="#second_page">List Item 2</a></li>
        <li><a href="#second_page">List Item 3</a></li>
        <li><a href="#second_page">List Item 4</a></li>
      </ul>
    </div>
  </div>
  <!-- 두 번째 페이지 -->
  <div data-role="page" id="second_page">
    <div data-role="header">
      <h1>Other Page</h1>
    </div>
    <div data-role="content">
      <p>Lorem ipsum dolor sit amet</p>
    </div>
  </div>
</body>
```



❖ 내부 페이지 이동

- 뒤로 가기 버튼
 - 페이지의 data-add-back-btn 속성을 true로 입력
 - 페이지 이동 시 슬라이드 효과도 자동 추가

코드 31-13 data-add-back-btn 속성

```
<body>
  <!-- 첫 번째 페이지 -->
  <div data-role="page" id="first_page">
    <!--생략-->
  </div>
  <!-- 두 번째 페이지 -->
  <div data-role="page" id="second_page" data-add-back-btn="true">
    <!--생략-->
  </div>
</body>
```

그림 31-12 페이지 이동



❖ jQuery Mobile 이 제공하는 페이지 이동방식

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

코드 31-14 data-transition 속성

```
<ul data-role="listview">
  <li><a href="#second_page" data-transition="slide">List Item 1</a></li>
  <li><a href="#second_page" data-transition="slideup">List Item 2</a></li>
  <li><a href="#second_page" data-transition="slidedown">List Item 3</a></li>
  <li><a href="#second_page" data-transition="pop">List Item 4</a></li>
  <li><a href="#second_page" data-transition="fade">List Item 5</a></li>
  <li><a href="#second_page" data-transition="flip">List Item 6</a></li>
</ul>
```



❖ 페이지 이동의 다른 예

- data-rel 속성을 dialog로 입력
 - 다이얼로그 형태로 페이지 전환
 - 역시 data-transition 속성 사용 가능

코드 31-15 data-rel 속성

```
<ul data-role="listview">
  <li><a href="#second_page" data-transition="slide">List Item 1</a></li>
  <li><a href="#second_page" data-transition="slideup">List Item 2</a></li>
  <li><a href="#second_page" data-transition="slidedown">List Item 3</a></li>
  <li><a href="#second_page" data-transition="pop">List Item 4</a></li>
  <li><a href="#second_page" data-transition="fade">List Item 5</a></li>
  <li><a href="#second_page" data-transition="flip">List Item 6</a></li>

  <li><a href="#second_page" data-rel="dialog">List Item 7</a></li>
  <li><a href="#second_page" data-rel="dialog" data-transition="pop">List Item
    8</a></li>
  <li><a href="#second_page" data-rel="dialog" data-transition="slidedown">
    List Item 9</a></li>
  <li><a href="#second_page" data-rel="dialog" data-transition="flip">List Item
    10</a></li>
</ul>
```

그림 31-13 jQuery Mobile 다이얼로그



❖ 리스트 태그의 data-role 속성 예제

코드 31-16 기본 리스트

```
<body>
  <div data-role="page">
    <div data-role="header">
      <h1>Mobile</h1>
    </div>
    <div data-role="content">
      <ul data-role="listview">
        <li><a href="#">List Item 1</a></li>
        <li><a href="#">List Item 2</a></li>
        <li><a href="#">List Item 3</a></li>
        <li><a href="#">List Item 4</a></li>
        <li><a href="#">List Item 5</a></li>
      </ul>
    </div>
  </div>
</body>
```



❖ jQuery Mobile에서의 구분자

- 리스트의 요소를 구분하는 구분자
- li 태그의 data-role 속성값을 list-divider로 입력

코드 31-17 구분자

```
<body>
  <div data-role="page">
    <div data-role="header">
      <h1>Mobile</h1>
    </div>
    <div data-role="content">
      <ul data-role="listview">
        <li data-role="list-divider">Divider</li>
        <li><a href="#">List Item 1</a></li>
        <li><a href="#">List Item 2</a></li>
        <li data-role="list-divider">Divider</li>
        <li><a href="#">List Item 3</a></li>
        <li><a href="#">List Item 4</a></li>
        <li><a href="#">List Item 5</a></li>
        <li><a href="#">List Item 6</a></li>
        <li data-role="list-divider">Divider</li>
        <li><a href="#">List Item 7</a></li>
        <li><a href="#">List Item 8</a></li>
        <li><a href="#">List Item 9</a></li>
        <li><a href="#">List Item 0</a></li>
      </ul>
    </div>
  </div>
</body>
```

그림 31-14 구분자



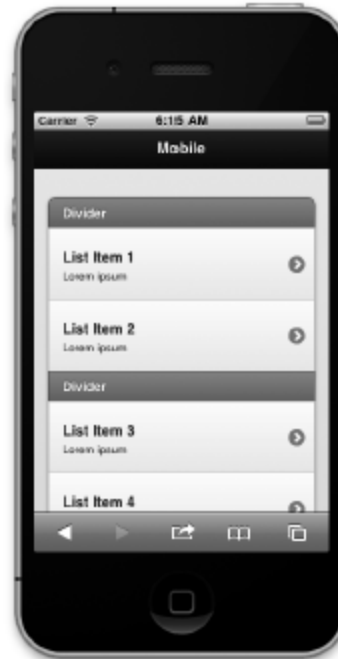
❖ 리스트를 하나의 그룹으로 구분하고 싶을 때

- data-inset 속성 true로 입력
 - 기본 설정은 false
- a 태그 안에 h3 태그와 p 태그
 - 자동으로 리스트의 요소에 제목과 내용 지정

코드 31-18 data-inset 속성

```
<ul data-role="listview" data-inset="true">
  <li data-role="list-divider">Divider</li>
  <li>
    <a href="#">
      <h3>List Item 1</h3>
      <p>Lorem ipsum</p>
    </a>
  </li>
</ul>
```

그림 31-15 data-inset 속성과 리스트 요소의 제목과 내용



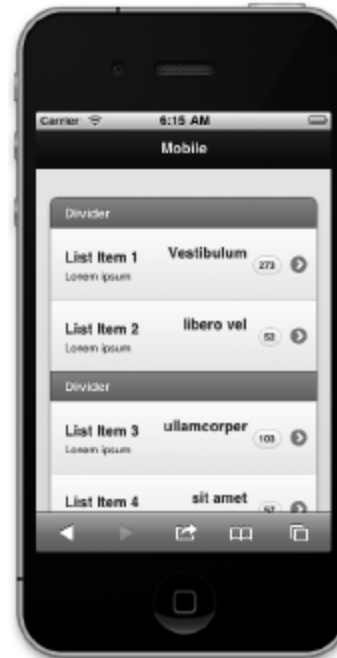
❖ 리스트 요소에 추가 설명과 카운터

- span 태그의 class 속성에 ui-li-aside와 ui-li-count 입력

코드 31-19 추가 설명과 카운터

```
<ul data-role="listview" data-inset="true">
  <li data-role="list-divider">Divider</li>
  <li>
    <a href="#">
      <h3>List Item 1</h3>
      <p>Lorem ipsum</p>
      <span class="ui-li-aside">Aside</span>
      <span class="ui-li-count">52</span>
    </a>
  </li>
</ul>
```

그림 31-16 리스트 요소의 추가 설명과 카운터



❖ 이미지 넣기

- 리스트에 이미지를 넣고 싶을 때 - img 태그
 - jQuery Mobile이 자동으로 이미지를 80×80의 크기로 변경해 삽입

코드 31-20 이미지와 아이콘

```
<ul data-role="listview" data-inset="true">
  <li data-role="list-divider">Divider</li>
  <li>
    <a href="#">
      
      <h3>List Item 1</h3>
      <p>Lorem ipsum</p>
      <span class="ui-li-count">52</span>
    </a>
    <a href="#" data-icon="search"></a>
  </li>
</ul>
```

그림 31-17 이미지와 버튼



❖ 리스트의 data-filter 속성

- 기본적으로 data-filter 속성은 false
- data-filter 속성을 true로 입력
 - 리스트의 요소를 검색하는 검색어 입력창 만들 수 있음

코드 31-21 data-filter 속성

```
<div data-role="content">
  <ul data-role="listview" data-inset="true" data-filter="true">
    <li data-role="list-divider">Divider</li>
    <li><a href="#">방화</a></li>
    <li><a href="#">개화산</a></li>
    <li><a href="#">김포공항</a></li>
    <li><a href="#">송정</a></li>
    <li><a href="#">마곡</a></li>
    <li><a href="#">발산</a></li>
    <li><a href="#">우정산</a></li>
    <li><a href="#">화곡</a></li>
    <li><a href="#">까치산</a></li>
  </ul>
</div>
```

그림 31-18 data-filter 속성을 사용한 검색창



❖ jQuery Mobile 자체 지원 아이콘

- data-icon 속성과 관련
- 자체적으로 그림 31-19와 같은 18개의 아이콘 지원

그림 31-19 jQuery Mobile의 아이콘



표 31-5 jQuery Mobile 아이콘의 이름

arrow-l	arrow-r	arrow-u
arrow-d	delete	plus
minus	check	gear
refresh	forward	back
grid	star	alert
info	home	search



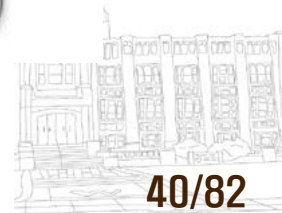
❖ jQuery Mobile 입력 양식

- 가장 간단한 형태는 type 속성
 - text와 password 갖는 input 태그
 - data-role 속성이 fieldcontain인 div 태그 사용시 모양 자동 조정

코드 31-22 기본 입력 양식

```
<div data-role="content">
  <div data-role="fieldcontain">
    <label for="textinput">Text Input</label>
    <input type="text" name="textinput" id="textinput" />
  </div>
  <div data-role="fieldcontain">
    <label for="passwordinput">Password Input</label>
    <input type="password" name="passwordinput" id="passwordinput" />
  </div>
</div>
```

그림 31-20 기본 입력 양식



❖ type 속성이 radio나 checkbox인 경우

- fieldset 태그의 data-role 속성을 controlgroup으로 입력
- label 태그의 for 속성을 정확하게 입력해야 예쁜 형태 형성

코드 31-23 체크 입력 양식

```
<div data-role="content">
  <div data-role="fieldcontain">
    <fieldset data-role="controlgroup">
      <legend>Check</legend>
      <label for="radioinput_1">Radio Input</label>
      <input type="radio" name="radioinput" id="radioinput_1" />
      <label for="radioinput_2">Radio Input</label>
      <input type="radio" name="radioinput" id="radioinput_2" />
    </fieldset>
  </div>
  <div data-role="fieldcontain">
    <fieldset data-role="controlgroup">
      <legend>Check</legend>
      <label for="checkboxinput_1">Checkbox Input</label>
      <input type="checkbox" name="checkboxinput_1" id="checkboxinput_1"/>
      <label for="checkboxinput_2">Checkbox Input</label>
      <input type="checkbox" name="checkboxinput_2" id="checkboxinput_2"/>
    </fieldset>
  </div>
</div>
```

그림 31-21 체크가 가능한 입력 양식



❖ 그 외의 입력 태그

■ select 태그

- 내부에 data-role 속성을 slider로 입력
- 2개의 option 태그를 넣으면 아이폰의 on & off 버튼과 같은 형태로 변경

코드 31-24 select 태그와 textarea 태그

```
<div data-role="content">
  <div data-role="fieldcontain">
    <label for="select">Select Input</label>
    <select id="select">
      <option>서울</option>
      <option>부산</option>
      <option>대구</option>
      <option>대전</option>
    </select>
  </div>
  <div data-role="fieldcontain">
    <label for="slider">Select slider</label>
    <select name="slider" id="slider" data-role="slider">
      <option value="off">Off</option>
      <option value="on">On</option>
    </select>
  </div>
  <div data-role="fieldcontain">
    <label for="textarea">Textarea Input</label>
    <textarea id="textarea"></textarea>
  </div>
</div>
```

그림 31-22 textarea 태그와 select 태그



❖ HTML5의 input 태그

- type 속성에 range나 search를 입력하면 그에 맞는 형태로 변화

코드 31-25 HTML5 입력양식 태그

```
<div data-role="content">
  <div data-role="fieldcontain">
    <label for="slider">Input slider</label>
    <input type="range" name="slider" id="slider" value="0" min="0" max="100" />
  </div>
  <div data-role="fieldcontain">
    <label for="search">Search Input</label>
    <input type="search" name="search" id="search" />
  </div>
</div>
```

그림 31-23 HTML5 입력 양식



❖ jQuery Mobile 아코디언

- 모바일 화면의 특성상 한 화면에 모든 내용을 담기 어려움
- 아코디언 형태의 위젯을 많이 사용
 - div 태그의 data-role 속성을 collapsible로 입력 – 기본은 열린 모습
 - 내부에 원하는 내용을 입력하면 아코디언 생성
 - 내부에 h1 태그를 입력하면 해당 h1 태그가 아코디언의 제목이 됨

코드 31-26 기본 아코디언

```
<div data-role="page">
  <div data-role="header"><h1>Mobile</h1></div>
  <div data-role="content">
    <div data-role="collapsible">
      <h1>Accordion</h1>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    </div>
    <div data-role="collapsible">
      <h1>Accordion</h1>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    </div>
    <div data-role="collapsible">
      <h1>Accordion</h1>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    </div>
  </div>
</div>
```

그림 31-24 jQuery Mobile 아코디언



❖ 아코디언의 열림/ 닫힘 상태를 조절하고 싶을 때

- data-collapsed 속성 사용
 - data-collapsed 속성은 기본적으로 false
- data-role 속성이 collapsible인 div 태그는 data-role 속성이 collapsible-set인 div 태그에 넣어주면 자동으로 그룹 형성

코드 31-27 아코디언 그룹

```
<div data-role="content">
  <div data-role="collapsible-set">
    <div data-role="collapsible">
      <h1>Accordion</h1>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    </div>
    <div data-role="collapsible">
      <h1>Accordion</h1>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    </div>
    <div data-role="collapsible">
      <h1>Accordion</h1>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    </div>
  </div>
</div>
```

그림 31-25 아코디언의 그룹화



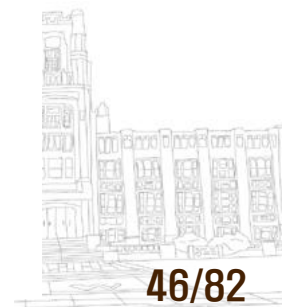
❖ jQuery Mobile 레이아웃

- div 태그의 class 속성에 ui-grid를 입력해 레이아웃 구성
 - ui-grid-a부터 ui-grid-d까지 입력 가능 (화면을 2~5등분)
 - ui-grid-a 사용해 레이아웃 구성
 - 내부 요소 클래스에 ui-block-a와 ui-block-b를 입력해 위치 지정

코드 31-28 ui-grid 클래스 (1)

```
<div data-role="content">
  <div class="ui-grid-a">
    <div class="ui-block-a"><h1>ui-block-a</h1></div>
    <div class="ui-block-b"><h1>ui-block-b</h1></div>
  </div>
</div>
```

그림 31-26 실행 결과



❖ jQuery Mobile 레이아웃 (2)

- 한 줄에 입력 가능
- 예제 코드 31-29처럼 ui-block 클래스 입력
 - 여러 줄에 요소를 구성 할 수 있음

코드 31-29 ui-grid 클래스 (2)

```
<div data-role="content">
  <div class="ui-grid-a">
    <div class="ui-block-a">h1</div>ui-block-a</div>
    <div class="ui-block-b">h1</div>ui-block-b</div>

    <div class="ui-block-a">h1</div>ui-block-a</div>
    <div class="ui-block-b">h1</div>ui-block-b</div>
  </div>
</div>
```

그림 31-27 실행 결과



❖ jQuery Mobile 레이아웃 (3)

- ui-grid-b 클래스는 화면 3등분
 - 내부에 ui-block 클래스 c까지 지정 가능
- 사용자의 화면 해상도에 따라 화면 구성이 깨질 수 있음
 - 4등분 이상은 되도록 사용하지 말 것

코드 31-30 ui-grid 클래스 (3)

```
<div data-role="content">
  <div class="ui-grid-b">
    <div class="ui-block-a">h1ui-block-a</h1></div>
    <div class="ui-block-b">h1ui-block-b</h1></div>
    <div class="ui-block-c">h1ui-block-c</h1></div>

    <div class="ui-block-a">h1ui-block-a</h1></div>
    <div class="ui-block-b">h1ui-block-b</h1></div>
    <div class="ui-block-c">h1ui-block-c</h1></div>
  </div>
</div>
```

그림 31-28 실행 결과



❖ 예제 코드 31-31]

- 레이아웃을 구성할 때 ui-grid를 하나의 구성 요소로 활용
- 원하는 곳에 버튼을 위치시키려고 ui-grid 부분적 사용

코드 31-31 ui-grid 클래스를 사용한 레이아웃 구성

```
<div data-role="content">
  <div data-role="fieldcontain">
    <label for="textinput">ID</label>
    <input type="text" name="id_input" id="id_input" />
  </div>
  <div data-role="fieldcontain">
    <label for="textinput">Password</label>
    <input type="text" name="password_input" id="password_input" />
  </div>
  <div class="ui-grid-a">
    <div class="ui-block-a"><input type="submit" value="submit" /></div>
    <div class="ui-block-b"><input type="reset" value="reset" /></div>
  </div>
</div>
```

그림 31-29 실행 결과



❖ 헤더와 푸터 – 합쳐서 툴바라 부름

- data-position 속성은 툴바의 위치 형태 지정
- data-position 속성은 inline이나 fixed 입력
 - 기본적으로 data-position 속성은 inlinedata-position 속성을 fixed로 사용
» 그림 31-30은 툴바 위치 고정 예

코드 31-32 data-position 속성

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

그림 31-30 실행 결과



❖ jQuery Mobile은 헤더에 기본적으로 두 개의 버튼 가능

- a 태그를 예제와 같이 입력 - 자동 버튼 생성
- h1 태그의 왼쪽과 오른쪽
 - 버튼이 자동으로 순서대로 지정 (위치 지정되지 않음)
 - 버튼의 위치를 명확하게 표시하고 싶을 때

– class 속성에 ui-btn-left나 ui-btn-right 입력

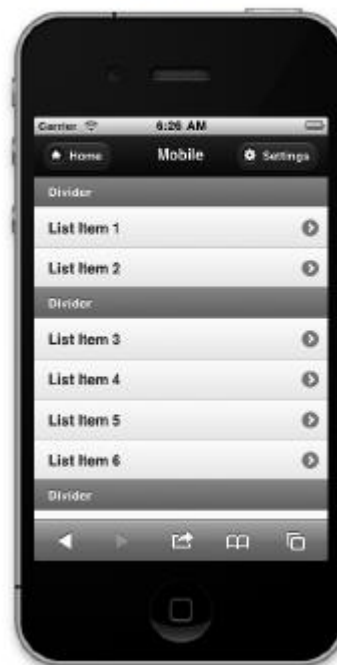
코드 31-33 툴바 버튼 생성

```
<div data-role="header" data-position="fixed">
  <a href="#" data-icon="home">Home</a>
  <h1>Mobile</h1>
  <a href="#" data-icon="gear">Settings</a>
</div>
```

코드 31-34 툴바 버튼의 클래스

```
<div data-role="header" data-position="fixed">
  <a href="#" data-icon="home" class="ui-btn-left">Home</a>
  <h1>Mobile</h1>
  <a href="#" data-icon="gear" class="ui-btn-right">Settings</a>
</div>
```

그림 31-31 실행 결과



❖ 푸터에는 버튼을 원하는 만큼 생성 가능

- 화면의 너비를 벗어나면 예쁘지 않으므로 세 개나 네 개가 가장 적당
- a 태그를 입력하면 버튼 생성

코드 31-35 툴바 버튼

```
<div data-role="footer" data-position="fixed">  
  <a href="#" data-icon="plus">Add</a>  
  <a href="#" data-icon="delete">Delete</a>  
  <a href="#" data-icon="refresh">Refresh</a>  
</div>
```

그림 31-32 실행 결과



❖ 내비게이션 바

- 탭 바 라고도 불림
- 헤더나 푸터에 div 태그 생성하고 data-role 속성 navbar로 입력
- 해당 div 태그 안에 ul 태그로 리스트 만들면, 내비게이션 바 생성

코드 31-36 내비게이션바

```
<div data-role="header" data-position="fixed">
  <a href="#" data-icon="home" class="ui-btn-left">Home</a>
  <h1>Mobile</h1>
  <a href="#" data-icon="gear" class="ui-btn-right">Settings</a>
  <div data-role="navbar">
    <ul>
      <li><a href="#">One</a></li>
      <li><a href="#">Two</a></li>
    </ul>
  </div>
</div>
```

그림 31-33 실행 결과



❖ 내비게이션 바 속의 아이콘

- data-icon 속성을 입력
 - 툴바 버튼과 마찬가지로 아이콘 넣을 수 있음

코드 31-37 내비게이션 바의 아이콘 설정

```
<div data-role="header" data-position="fixed">
  <a href="#" data-icon="home" class="ui-btn-left">Home</a>
  <h1>Mobile</h1>
  <a href="#" data-icon="gear" class="ui-btn-right">Settings</a>
  <div data-role="navbar">
    <ul>
      <li><a href="#" data-icon="grid">One</a></li>
      <li><a href="#" data-icon="star">Two</a></li>
      <li><a href="#" data-icon="alert">Three</a></li>
      <li><a href="#" data-icon="info">Four</a></li>
      <li><a href="#" data-icon="gear">Five</a></li>
    </ul>
  </div>
</div>
```

그림 31-34 실행 결과



❖ 내비게이션 바 활성화

- class 속성을 ui-btnactive

코드 31-38 내비게이션바의 상태 클래스

```
<div data-role="navbar">
  <ul>
    <li><a href="#" data-icon="grid" class="ui-btn-active">One</a></li>
    <li><a href="#" data-icon="star">Two</a></li>
    <li><a href="#" data-icon="alert">Three</a></li>
    <li><a href="#" data-icon="info">Four</a></li>
    <li><a href="#" data-icon="gear">Five</a></li>
  </ul>
</div>
```



❖ jQuery Mobile은 기본적으로 테마 가짐

- 페이지는 검정색 테마고 리스트 뷰에 들어가는 구성 요소들은 파란색
- jQuery Mobile은 기본적으로 다섯 종류 (a, b, c, d, e) 의 테마 지원
- 태그에 data-theme 속성 지정하면 테마 줄 수 있음
- 페이지에 data-theme 속성 입력하면 페이지 전체에 테마 적용

코드 31-39 data-theme 속성

```
<body>
  <div data-role="page" data-theme="c">
    <!-- 생략 -->
  </div>
</body>
```

그림 31-35 실행 결과



❖ jQuery Mobile 테마

- 가장 기본적인 테마는 a 테마
 - 테마를 적용하지 않았을 때 적용되던 검정색 테마가 a 테마
 - b 테마는 파란색, c 테마는 회색, d 테마는 흰색, e 테마는 노란색
- 각 요소에 테마를 적용해보는 예제

코드 31-40 각각의 요소에 테마 적용

```
<body>
  <div data-role="page" data-theme="d">
    <div data-role="header">
      <a href="#" data-icon="gear" class="ui-btn-left">Settings</a>
      <h1>Mobile</h1>
      <a href="#" data-icon="gear" class="ui-btn-right">Settings</a>
    <div data-role="navbar">
      <ul>
        <li><a href="#" data-icon="home" data-theme="b">One</a></li>
        <li><a href="#" data-icon="gear" data-theme="c">Two</a></li>
        <li><a href="#" data-icon="grid" data-theme="d">Two</a></li>
        <li><a href="#" data-icon="star" data-theme="e">Two</a></li>
      </ul>
    </div>
  </div>
```

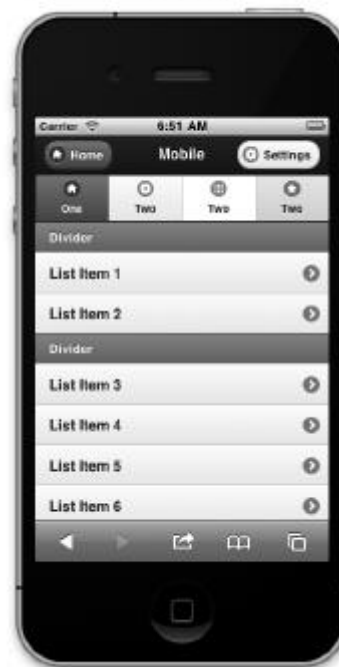


❖ jQuery Mobile 테마

- 각 요소에 테마를 적용해보는 예제 (2)

```
<div data-role="content">
  <ul data-role="listview">
    <li data-role="list-divider">Divider</li>
    <li><a href="#">List Item 1</a></li>
    <li><a href="#">List Item 2</a></li>
    <li data-role="list-divider">Divider</li>
    <li><a href="#">List Item 3</a></li>
    <li><a href="#">List Item 4</a></li>
    <li><a href="#">List Item 5</a></li>
    <li><a href="#">List Item 6</a></li>
    <li data-role="list-divider">Divider</li>
    <li><a href="#">List Item 7</a></li>
    <li><a href="#">List Item 8</a></li>
    <li><a href="#">List Item 9</a></li>
    <li><a href="#">List Item 0</a></li>
  </ul>
</div>
</div>
</body>
```

그림 31-36 실행 결과



❖ 사용자 지정 테마

- jQuery Mobile Theme Roller를 사용해 만들 수 있음
- 각 부분에 해당하는 색상을 설정하여 테마를 만들고 다운로드
- 기존의 css 파일 대신 다운받은 파일을 추가하고 datatheme 속성 지정

그림 31-37 jQuery Theme Roller(<http://jquerymobile.com/themeroller/>)



❖ 다른 파일의 페이지로 이동하는 방법

- jQuery Mobile은 a 태그를 눌러 이동할 대상 페이지를 세 가지로 구분
 - 내부 페이지: 같은 HTML 페이지 안에 존재하는 페이지
 - 외부 페이지: 다른 도메인에 존재하는 페이지
 - 로컬 페이지: 같은 도메인에 존재하는 페이지
- a 태그 클릭해 이동
 - 페이지의 종류에 따라 다른 형태로 페이지 이동
- 예제 코드 생성

그림 31-37 폴더 구성



❖ MainPage.htm 파일

- jQuery Mobile을 사용하기 위한 구성 요소 추가해 페이지 생성
 - content 내부의 첫 번째 링크는 외부 페이지로 링크
 - 두 번째 링크는 내부 페이지로 링크

코드 31-41 MainPage.htm: HTML 페이지 구성

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0,
    user-scalable=no"/>
  <link rel="stylesheet" href="http://code.jquery.com/mobile/1.0b1/jquery.
    mobile-1.0b1.min.css"/>
  <script src="http://code.jquery.com/jquery-1.6.2.min.js"></script>
  <script src="http://code.jquery.com/mobile/1.0b1/jquery.mobile-1.0b1.min.js"></script>
</head>
<body>
  <div data-role="page">
    <div data-role="header">
      <h1>Mobile</h1>
    </div>
    <div data-role="content">
      <a href="http://hanb.co.kr" data-role="button">External Link</a>
      <a href="OtherPage.htm" data-role="button">Local Link</a>
    </div>
  </div>
</body>
</html>
```



❖ 두 번째 페이지 구성

- 이동됐음을 화면에 표시하는 용도
- head 태그 안에 jQuery Mobile을 위한 구성을 별도로 하지 않음

코드 31-42 OtherPage.htm : HTML 페이지 구성

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
  <div data-role="page" data-add-back-btn="true">
    <div data-role="header">
      <h1>Mobile</h1>
    </div>
    <div data-role="content">
      <h1>Hello Second Page</h1>
    </div>
  </div>
</body>
</html>
```



❖ 실행 결과

- 실행하고 각 버튼을 누르면 그림 31-38처럼 페이지 전환
 - 외부 페이지는 일반 링크 취급
 - 로컬 페이지는 자동으로 Ajax 수행
 - 대상 HTML 페이지의 body 태그를 현재 HTML페이지에 추가 후 이동

그림 31-38 jQuery Mobile의 페이지 이동 방식



❖ 실행 결과

- Ajax를 사용하고 싶지 않다면?
 - rel 속성을 external로 입력
 - `Local Link`

그림 31-39 로컬 페이지의 이동



❖ 메일 기능과 전화 기능

- a 태그 사용
- a 태그의 href 속성에 "mailto:메일" 또는 "tel:전화번호" 입력
 - 스마트폰의 내장 메일 기능이나 전화 기능 이용 가능

코드 31-43 스마트폰의 메일 또는 전화 사용

```
<a href="mailto:rintiantta@naver.com" data-role="button">Mail Link</a>
<a href="mailto:rintiantta@naver.com?subject=Hello&body=World..!"
  data-role="button">Mail Link</a>
<a href="tel:010-0000-0000" data-role="button">Phone Link</a>
```

표 31-6 mailto의 속성

속성 이름	설명
cc	참조
bcc	숨은 참조
subject	제목
body	내용물

그림 31-40 실행 결과



❖ mobileinit 이벤트

- jQuery Mobile 페이지가 형성되면서 실행
- 주의할 점
 - jQuery Mobile과 관련된 **자바스크립트 파일을 읽을 때** 실행하는 이벤트
- JQueryMobile.js 파일을 추가하기 전 사용

코드 31-44 mobileinit 이벤트

```
<script src="http://code.jquery.com/jquery-1.6.2.min.js">/script>
<script>
    $(document).bind('mobileinit', function () {
        alert('Mobile Init..!');
    });
</script>
<script src="http://code.jquery.com/mobile/1.0b1/jquery.mobile-1.0b1.min.js">
</script>
```

코드 31-45 뒤로 가기 버튼 자동 추가

```
<script>
    $(document).bind('mobileinit', function () {
        $.mobile.page.prototype.options.addBackBtn = true;
    });
</script>
```



❖ jQuery Mobile의 페이지 관련 이벤트

표 31-7 jQuery Mobile의 페이지 관련 이벤트

이벤트 이름	설명
pagebeforecreate	페이지를 생성하기 바로 전에 실행합니다.
pagecreate	페이지가 생성하고 실행합니다.
pagebeforechange	페이지를 이동하기 바로 전에 실행합니다.
pagechange	페이지를 이동하고 실행합니다.
pagechangefailed	페이지 이동이 실패했을 때 실행합니다.



❖ jQuery Mobile의 페이지 관련 이벤트 예제

코드 31-46 페이지 관련 이벤트

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0,
    maximum-scale=1.0, minimum-scale=1.0, user-scalable=no" />
  <link href="http://code.jquery.com/mobile/latest/jquery.mobile.min.
    css" rel="stylesheet" type="text/css"/>
  <script src="http://code.jquery.com/jquery-1.6.2.min.js" />
  <script>
    $(document).bind('mobileinit', function () {
      alert('MOBILE INIT');
    });
    $(document).bind({
pagebeforecreate: function () { alert('PAGE BEFORE CREATE'); },
pagecreate: function () { alert('PAGE CREATE'); },
pagebeforechange: function () { alert('PAGE BEFORE CHANGE'); },
pagechange: function () { alert('PAGE CHANGE'); }
    });
  </script>
  <script src="http://code.jquery.com/mobile/latest/jquery.mobile.min.js" />
</head>
```



❖ jQuery Mobile의 페이지 관련 이벤트 예제 (2)

```
<body>
  <!-- 첫 번째 페이지 -->
  <div data-role="page">
    <div data-role="header">
      <h1>FIRST</h1>
    </div>
    <div data-role="content">
      <a href="#second_page" data-role="button">To Second Page</a>
    </div>
  </div>
  <!-- 두 번째 페이지 -->
  <div data-role="page" id="second_page" data-add-back-btn="true">
    <div data-role="header">
      <h1>SECOND</h1>
    </div>
    <div data-role="content"></div>
  </div>
</body>
</html>
```



❖ jQuery Mobile에서 자주 사용하는 메서드

표 31-8 jQuery Mobile의 메서드

메서드 이름	설명
<code>\$.mobile.changePage(url, options)</code>	페이지를 이동합니다.
<code>\$.mobile.loadPage(url, options)</code>	페이지를 불러옵니다.
<code>\$.mobile.showPageLoadingMsg()</code>	로딩 메시지를 출력합니다.
<code>\$.mobile.hidePageLoadingMsg()</code>	로딩 메시지를 제거합니다.



❖ FirstPage.htm 페이지

- 페이지를 만들고 페이지를 이동할 수 있는 버튼 생성
- 메서드를 사용해 페이지 이동
 - a 태그의 기본 이벤트를 제거

코드 31-47 FirstPage.htm :body 태그 구성

```
<body>
  <div data-role="page">
    <div data-role="header">
      <h1>MOBILE</h1>
    </div>
    <div data-role="content">
      <h1>Here is First PAGE</h1>
      <a data-role="button" href="SecondPage.htm">Move To Second PAGE</a>
    </div>
  </div>
</body>
```



❖ SecondPage.htm

- 두 번째 페이지라는 것만 표시

코드 31-48 SecondPage.htm: body 태그 구성

```
<body>
  <div data-role="page">
    <div data-role="header">
      <h1>MOBILE</h1>
    </div>
    <div data-role="content">
      <h1>Here is Second PAGE</h1>
    </div>
  </div>
</body>
```



❖ FirstPage.htm 페이지에 자바스크립트 입력

- document 객체의 pagecreate 이벤트에 연결
- pagecreate 이벤트는 jQuery Mobile이 지원하는 이벤트
 - 페이지가 모두 만들어졌을 때 실행
- 첫 번째 페이지가 완성되면 a 태그에 이벤트 연결
- SecondPage.htm을 매개 변수로 loadPage() 메서드 실행

코드 31-49 FirstPage.htm: \$.mobile.loadPage() 메서드

```
<script>
    $(document).bind('pagecreate', function () {
        $('a').click(function () {
            // 변수를 선언합니다.
            var href = $(this).attr('href');
            // 페이지를 이동합니다.
            $.mobile.loadPage(href);
            // 기본 이벤트를 제거합니다.
            return false;
        });
    });
</script>
```

그림 31-42 아무 반응 없는 페이지



❖ loadPage() 메서드

- 페이지를 직접적으로 이동하지 않고 불러오기만 함
- jQuery Mobile은 기본적으로 Ajax 사용해 페이지 이동
 - 페이지를 현재 페이지에 Ajax 사용해 로드 후
 - 현재 페이지 내에서 페이지 이동
- loadPage() 메서드 - Ajax를 사용해 로드하는 부분까지만 실행

그림 31-43 페이지는 이동하지 않지만 추가는 합니다.

```
<body class="ui-mobile-viewport">
  <div data-role="page" class="ui-page ui-body-c ui-page-active" style="min-height: 597px; ">...</div>
  <div class="ui-loader ui-body-a ui-corner-all" style="top: 298.5px; ">...</div>
  <div data-role="page" data-url="/jQueryPageTransition/SecondPage.htm" class="ui-page ui-body-c">...</div>
</body>
```



❖ changePage() 메서드

- Ajax를 사용하여 페이지를 로드한 후 이동까지 수행

코드 31-50 FirstPage.htm: \$.mobile.changePage() 메서드

```
<script>
    $(document).bind('pagecreate', function () {
        $('a').click(function () {
            // 변수를 선언합니다.
            var href = $(this).attr('href');
            // 페이지를 이동합니다.
            $.mobile.changePage(href);
            // 기본 이벤트를 제거합니다.
            return false;
        });
    });
</script>
```



❖ showPageLoadingMsg() 메서드

- 페이지 이동시 jQuery Mobile 페이지는 자동으로 로딩 메시지 출력
 - 짧으면 못 보는 경우도 있음
- 로딩 메시지를 강제로 보여주고 싶을 때

코드 31-51 FirstPage.htm: \$.mobile.showPageLoadingMsg() 메서드

```
<script>
    $(document).bind('pagecreate', function () {
        $('a').click(function () {
            // 로딩 메시지를 출력합니다.
            $.mobile.showPageLoadingMsg()
            // 기본 이벤트를 제거합니다.
            return false;
        });
    });
</script>
```



❖ showPageLoadingMsg() 메서드 (2)

- 로딩 메시지에 원하는 글자 입력
 - loadingMessage 속성에 문자열 입력

코드 31-52 FirstPage.htm : 로딩 메시지 지정

```
<script>
    $(document).bind('pagecreate', function () {
        $('a').click(function () {
            // 로딩 메시지를 출력합니다.
            $.mobile.loadingMessage = '이동합니다.';
            $.mobile.showPageLoadingMsg()
            // 기본 이벤트를 제거합니다.
            return false;
        });
    });
</script>
```

그림 31-44 실행 결과



❖ jQuery Mobile은 플러그인의 집합체

- jQuery UI 플러그인을 적용하는 것처럼 메서드 호출해 UI 구성
 - 원하지 않는 문제 야기 가능

코드 31-53 body 태그 구성

```
<body>
  <div data-role="page">
    <div data-role="header">
      <h1>MOBILE</h1>
    </div>
    <div data-role="content">
      <button>ADD BUTTON</button>
    </div>
  </div>
</body>
```

그림 31-45 실행 결과



코드 31-54 문서 객체 생성

```
<script>
  $(document).bind('pagecreate', function () {
    $('button').click(function () {
      // 문서 객체를 추가합니다.
      $('<button>ADD</button>').appendTo('div[data-role=content]');
    });
  });
</script>
```



❖ 동적으로 생성한 문서 객체

- UI를 강제로 적용해야 함
 - jQuery UI는 각 위젯 객체의 메서드 호출
 - jQuery Mobile은 해당 태그를 감싼 문서 객체의 create 이벤트를 강제 호출

코드 31-55 create 이벤트 강제 호출

```
<script>
$(document).bind('pagecreate', function () {
    $('button').click(function () {
        // 문서 객체를 추가합니다.
        $('<button>ADD</button>').appendTo('div[data-role=content]');

        // create 이벤트를 강제로 호출합니다.
        $('div[data-role=content]').trigger('create');
    });
});
</script>
```

그림 31-46 실행 결과



❖ 동적으로 문서 객체를 생성할 경우

- create 이벤트 실행
- 리스트에 요소를 추가한 경우
 - refresh() 메서드 사용
- 헤더의 오른쪽에 추가용 버튼 생성

코드 31-56 body 태그 구성

```
<body>
  <div data-role="page">
    <div data-role="header">
      <h1>MOBILE</h1>
      <a data-icon="plus" class="ui-btn-right">ADD</a>
    </div>
    <div data-role="content">
      <ul data-role="listview">
        <li><a href="#">List Item</a></li>
      </ul>
    </div>
  </div>
</body>
```



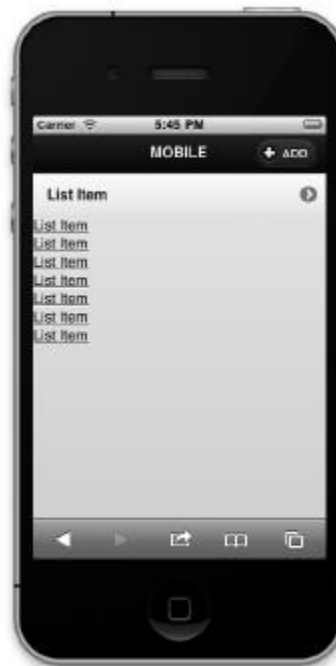
❖ 동적으로 문서 객체를 생성할 경우

- 리스트 요소도 동적 추가

코드 31-57 동적으로 리스트 요소 추가

```
<script>
$(document).bind('pagecreate', function () {
    $('a[class=ui-btn-right]').click(function () {
        // 문서 객체를 추가합니다.
        $('<li><a href="#">List Item</a></li>').appendTo('ul');
        // 기본 이벤트를 제거합니다.
        return false;
    });
});
</script>
```

그림 31-47 실행 결과



❖ listview() 플러그인의 refresh() 메서드

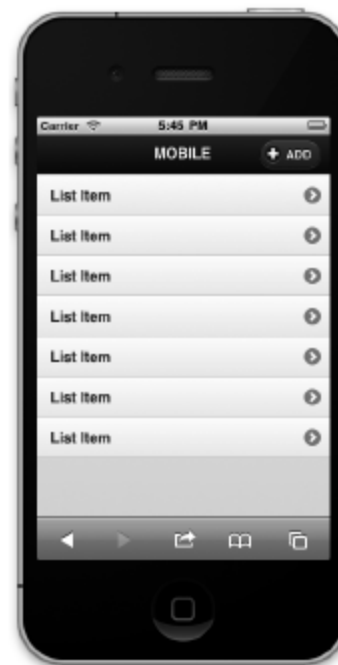
- 리스트의 요소 추가할 때 꼭 실행
 - jQuery Mobile의 리스트 모양 형성

코드 31-58 Listview 플러그인의 refresh() 메서드

```
<script>
    $(document).bind('pagecreate', function () {
        $('a[class=ui-btn-right]').click(function () {
            // 문서 객체를 추가합니다.
            $('<li><a href="#">List Item</a></li>').appendTo('ul');
            $('ul').listview('refresh');

            // 기본 이벤트를 제거합니다.
            return false;
        });
    });
</script>
```

그림 31-48 실행 결과





Thank You !

모던 웹을 위한 Javascript jQuery 입문