

# Markup General Guideline

Ver.0.2

Uracle UX Team

# Index

## ● Introduce

1. 하이브리드앱이란?
2. 하이브리드앱 Markup Guideline

## ● Device General Guide

1. 기본구성
2. HTML5
3. CSS3

## ● 하이브리드앱 Markup Basic Interaction

1. 서비스 디바이스 별 최적 해상도
2. 기본 선언 및 폴더구조
3. 생산성 향상을 위한 Tip
4. 앱/웹 접근성 준수

## ● Introduce

1. 하이브리드란?
2. Markup Process

## 1. 하이브리드앱이란?

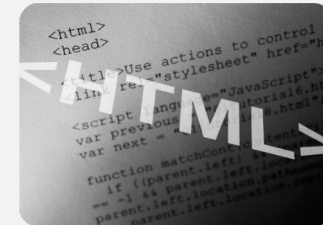
HTML과 같은 웹표준 언어로 한 번 개발한 애플리케이션을  
Android, iOS 등 여러 스마트폰 OS에 통합 적용할 수 있는 Platform 입니다.

### 다양한 OS와 디바이스 지원



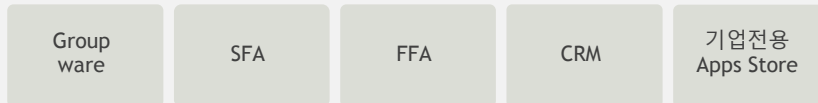
- OS 종류나 화면 크기 등에 관계 없이 '원소스 멀티 유즈 (One Source Multi Use)'가 가능하기 때문에 **비용을 절감하고 시간을 줄여 줍니다.**

### HTML 언어 채택



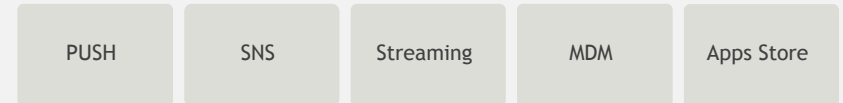
- 개발언어로는 개발자 및 기업IT 담당자들에게 익숙한 HTML을 채택해 **개발 및 유지관리를 수월하게 할 수 있습니다.**

### 기업 모빌리티 환경 구축



- Legacy 시스템과 연동하여 **기업에서 우선적으로 필요로 하는 Groupware, SFA, FFA, CRM 등의 애플리케이션을** 빠르게 구축할 수 있는 환경을 제공합니다.

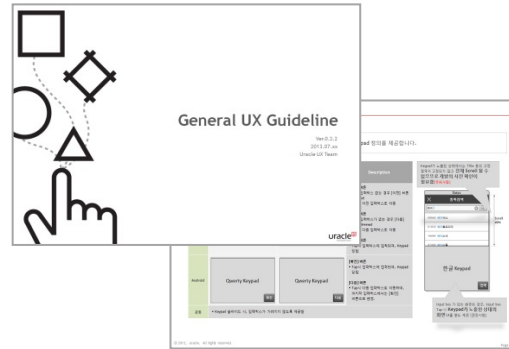
### 부가 Platform/Contents



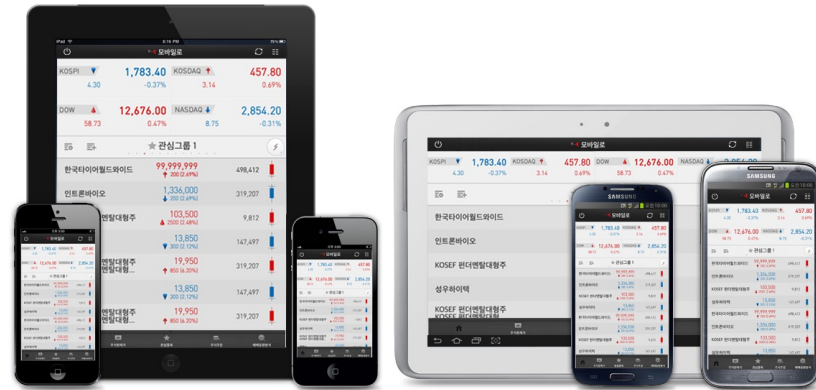
- 푸시 알림 서비스, 동영상 Platform, 소셜 네트워크 서비스(SNS) 등 **다양한 부가 Platform과 관리 Platform까지 통합적으로 지원합니다.**

## 2. 하이브리드 앱 Markup Process

하이브리드 로 개발되어 Multi-Device에서 실행되는 모바일 애플리케이션의 최적화 된 Markup을 위한 Guideline을 제공합니다.



\*\* General UX Guideline



### [Markup Process]

Service Device & model & OS

Service Device mode=portrait / landscape

Markup Layout & Guide

서비스 디바이스 확인  
(mobile & Tablet)

최소 OS,  
최소 기종 확인  
(Android & iOS)

서비스 모드  
(가로 & 세로모드)

확정

공통 Layout 및  
Guide설계

## 3. 하이브리드앱 Markup Process

### 서비스 디바이스 확인(mobile & Tablet)



- Mobile 과 Tablet을 하나의 소스로 동시 지원하는 지 확인해야 합니다.
- 동시 지원일 경우 디자인이 별도로 구성되는지 동일하게 구성되는지 확인해야 합니다.

### 서비스 최소 OS, 최소 기종 확인 (Android & iOS)



- 각 디바이스 별 OS 최소 버전을 확인해야 합니다. (예 : Android 4.2 이상 , iOS 5.0이상 )
- 디바이스의 최소 기종을 파악해야 합니다. (예: S3 이상, iPhone4 이상 등)

### 디바이스 별 서비스 모드 (가로 & 세로모드)



- Mobile의 경우에는 기본적으로 세로모드만 지원하도록 합니다.
- Tablet 혹은 iPad 인 경우에는 가로 or 세로 모드를 모두 지원하는 지 둘 중 한가지 모드만 지원하는지 여부를 반드시 확인 하여야 합니다.

### 공통 Layout 및 Guide설계



- 기본적으로 공통 Layout 및 Common Guide 부분을 먼저 설계하여, 다수의 작업자가 함께 진행할 때 이슈가 없도록 합니다.

# Device General Guide

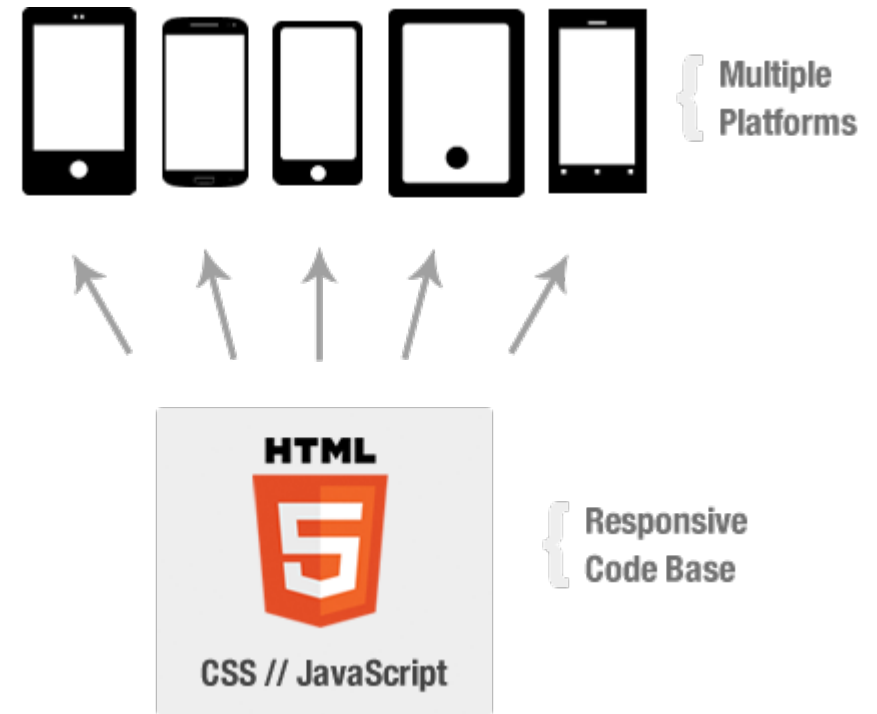
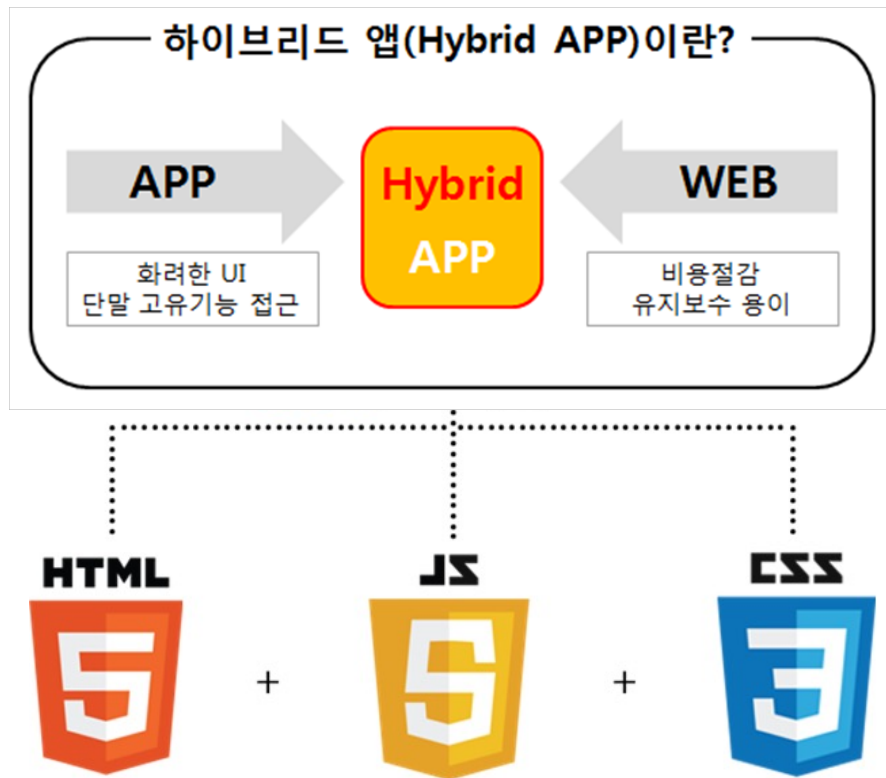
## ● Device General Guide

1. 기본구성
2. HTML5
3. CSS3



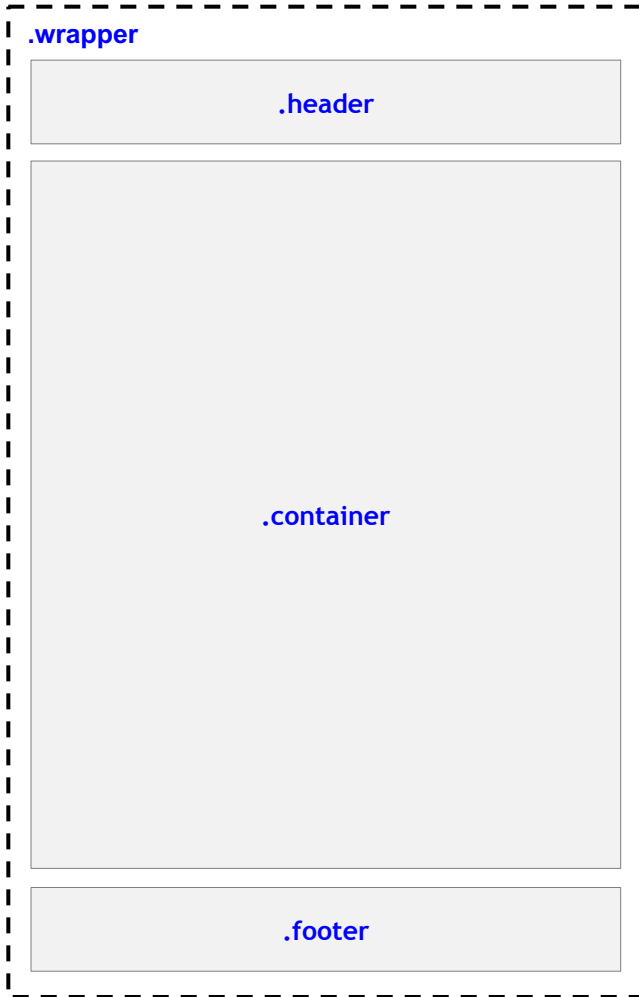
### 1. 기본구성

HTML5, CSS3, JavaScript 를 이용하여 하이브리드 앱을 구현할 수 있습니다.



## 2. HTML5

HTML5 구성 시 공통으로 지켜야 할 선언과 환경은 다음과 같다.



```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no">
  <meta name="format-detection" content="telephone=no, address=no, email=no">
  <title>파일제목</title>
  <link rel="stylesheet" href="../css/morpheus_common.css">
</head>
<body>

<div class="sidemenu">
  사이드메뉴 들어가는 자리
</div>

<div class="wrapper">
  <!-- header -->
  <div class="header">
    상단 영역
  </div>

  <!-- container -->
  <div class="container">
    컨텐츠 영역
  </div>

  <!-- footer -->
  <div class="footer">
    하단 영역
  </div>
</div>

</body>
</html>
```

### 2. HTML5

```
<meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no">
```

#### • viewport

Multi-Device 화면에 적절하게 콘텐츠를 표시하기 위해 viewport를 위와 같이 합니다.

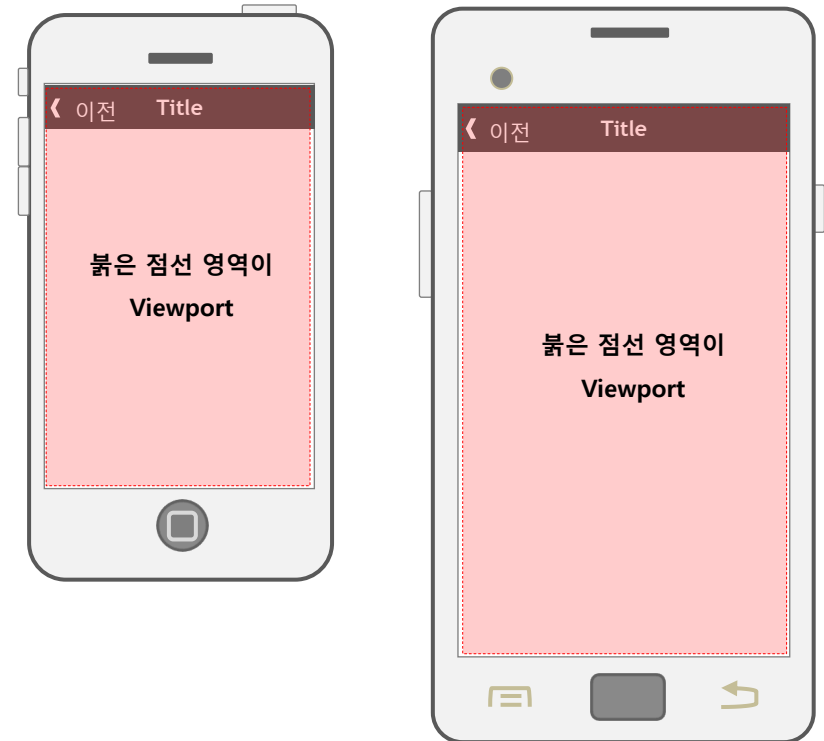
Width는 device-width로 선언하여 각 디바이스의 화면 해상도에 맞게 사이즈를 지정하고, initial-scale은 1로 선언하여 최초 스케일을 확대/축소 없는 상태로 지정하고, user-scalable은 no로 선언하여 사용자에게 의해서 화면이 확대/축소되는 것을 방지합니다.

위 세가지 속성 외, **minimum-scale, maximum-scale, target-densitydpi** 등의 속성은 **사용하지 않습니다**. 특히, target-densitydpi 속성은 Android에서만 유용한 속성이고, WebKit기반 브라우저에서는 향후 제거될 속성이니 상위 호환성을 위해서 사용하지 않습니다.

#### • 불필요한 속성

HTML5에서는 link, style, script 엘리먼트에 type 속성을 사용하지 않아도, 기본값인 text/css, text/javascript로 선언이 되기 때문에, 굳이 선언하지 않아도 됩니다.

그리고, 폼 요소의 속성 중 checked와 disabled도, 굳이 checked="checked" disabled="disabled"와 같이 길게 선언하지 않아도 HTML5에서는 checked, disabled로 간결하게 선언해도 무방하니, 코드를 최대한 간결하게 작성/유지하도록 합니다.



## 2. HTML5

### 시멘틱 마크업이란?

컨텐츠의 의미에 맞는 적절한 엘리먼트를 사용하여 의미 있는 HTML문서가 되도록 마크업 하는 것을 말합니다.

#### • 구조(HTML), 표현(CSS), 동작(JS)을 분리

위와 같이 구조와 표현 그리고 동작을 혼합하여 작성하면, 소스가 길어지고 읽기가 불편해집니다. 또한 유지/보수 측면에서 관리가 힘들어집니다. 따라서, 구조와 표현 그리고 동작을 최대한 분리하여 소스를 작성하도록 합니다.

즉, inline, internal 방식을 지양하고, external 방식으로 작성/관리 합니다.

간단히 말하면, 스타일은 CSS파일로 작성하고, 자바스크립트는 JS파일로 작성하여 다음과 같이 불러옵니다.

잘못된 예	올바른 예
<code>&lt;div style="font-size:20px;"&gt;구조와 표현을 혼합한 경우&lt;/div&gt;</code>	<code>&lt;link rel="stylesheet" href="../css/파일명.css"&gt;</code>
<code>&lt;div onclick="open();"&gt;구조와 동작을 혼합한 경우&lt;/div&gt;</code>	<code>&lt;script src="../js/파일명.js"&gt; &lt;/script&gt;</code>

#### • HTML5기능 사용 가능 여부 확인

HTML5, CSS3의 모든 기능을 아직은 모든 브라우저에서 지원을 하지는 못합니다. 따라서, 사용하려는 HTML5 기능이 iOS 또는 Android에서 사용이 가능한지를 우선 확인해야 합니다. CSS3, JavaScript도 다음 사이트에서 확인이 가능합니다.

<http://caniuse.com/>

#### • Script 위치

페이지 렌더링 성능을 향상을 위해서 script 엘리먼트는 </body> 바로 앞에 위치시키는 것이 좋습니다.

<head>엘리먼트 내에 위치시키거나 <body>엘리먼트의 중간 중간에 산발적으로 위치시키면 페이지 렌더링 성능이 떨어지게 됩니다.

#### • input 엘리먼트의 다양한 type속성

Device OS(iOS, Android)에 따라 그리고 같은 Device OS라도 버전에 따라 HTML5가 가지는 input의 속성에 대해서 지원여부와 형태가 다르니, 프로젝트 초기에 애플리케이션의 대상 Device OS 버전을 미리 정확히 체크한 후에 마크업을 시작하는 것이 좋습니다.

HTML5에서는 input 엘리먼트에 아래와 같이 새로운 type 속성들이 추가 되었습니다

<code>&lt;input type="search" /&gt;</code>	<code>&lt;input type="week" /&gt;</code>
<code>&lt;input type="tel" /&gt;</code>	<code>&lt;input type="time" /&gt;</code>
<code>&lt;input type="url" /&gt;</code>	<code>&lt;input type="datetime-local" /&gt;</code>
<code>&lt;input type="email" /&gt;</code>	<code>&lt;input type="number" /&gt;</code>
<code>&lt;input type="datetime" /&gt;</code>	<code>&lt;input type="range" /&gt;</code>
<code>&lt;input type="date" /&gt;</code>	<code>&lt;input type="color" /&gt;</code>
<code>&lt;input type="month" /&gt;</code>	

### 3. CSS3

기본적으로 제공되는 CSS는 morpheus\_reset.css, morpheus\_common.css, morpheus\_style.css로 구성되어 있습니다.

- morpheus-reset.css

**morpheus\_reset.css** 을 사용하여 CSS를 초기화하고 있습니다.

CSS초기화(CSS reset) 하는 방법에는 여러 가지가 있지만, margin:0;padding:0;과 같은 식의 초기화보다는, 브라우저마다 조금씩 다르게 랜더링하는 차이점을 없애고, 모든 브라우저에서 같은 값(value)으로 랜더링하는 방식으로 초기화하는 방식으로 구성되어 있습니다.

본 morpheus\_reset.css는 일반 사용자는 절대 수정하거나 삭제하지 않아야 합니다.

- morpheus-common.css

**morpheus\_common.css** 에서 @import를 이용하여 morpheus\_reset.css 선언하며, 그 외에는 @import는 사용하지 않습니다.

@import를 사용하면, 브라우저가 CSS파일을 병렬로 읽지 못하고, 직렬로 차례대로 읽기 때문에 랜더링 성능에 좋지 않기 때문입니다.

또한 morpheus\_common.css 에는 공통으로 사용되는 Layout 구조나 미디어쿼리, 공통 class를 선언하는 것이 좋습니다.

- morpheus-style.css

각 화면 별로 다르게 적용되는 속성이나, 다수의 작업자들이 프로젝트를 진행할 경우 개별로 사용할 수 있도록 선언된 css 입니다.

화면 별로 개별 id 값을 주어 상속으로 선언하는 것이 좋습니다.

```
<link rel="stylesheet" href="../css/morpheus-common.css"><!-- 공통 CSS -->
<link rel="stylesheet" href="../css/morpheus-style.css"><!-- 화면 별 CSS -->
```

- 미디어 쿼리

다양한 Device의 해상도에 맞게 화면을 보여줄 수 있도록 작업하기 위해 사용하는 제작 방식입니다.

- **Device 확대모드 지원** : 작은 모바일 화면과 그보다 큰 태블릿 화면에서 동일한 형태의 화면이 찌그러짐이 없이 보여지도록 하는 방식
- **Device 반응형 지원** : 동일한 콘텐츠를 작은 모바일 화면과 그보다 큰 태블릿 화면에서 다른 구조로 보여지게 화면 구성하는 방식

다음 사이트를 이용하면 현재 접속한 디바이스에 해당하는 미디어쿼리 정보를 손쉽게 알아낼 수 있습니다.

<http://atmedia.info/>

### 3. CSS3 단위

기본적으로 제공되는 CSS에서 사용하는 단위는 rem 입니다.

#### • rem

CSS3 font-size 속성 값으로 새로운 상대적 단위 'rem'을 사용할 수 있게 되었습니다.

CSS3의 글꼴 크기 값으로 새로 등장한 'rem' 단위는 root em 이라는 의미입니다. HTML 문서에서 root 요소는 <html> 요소를 의미하고 html 요소에 지정된 글꼴 크기로부터 선택된 요소의 글꼴 크기를 상대적으로 결정하는 것이 바로 rem 단위의 핵심입니다.

#### • 'em' vs 'rem'

- **em** : em단위는 부모로부터 글꼴 크기를 물려받기 때문에 같은 값을 지니더라도 노드가 깊어질수록 글꼴 크기가 기하 급수적으로 크거나 작아집니다. 이런 방식의 문제는 부모 또는 조상 노드 가운데 어떤 요소의 글꼴 사이즈 값을 변경할 때 모든 자식과 자손 요소도 그 영향을 받기 때문에 글꼴 사이즈를 예측하기 어렵고 계산하기 복잡하다는 치명적인 문제가 있습니다.
- **rem** : rem 단위는 부모 아닌 시조(root=html)로부터 글꼴 크기를 물려받기 때문에 html 요소에 기본 글꼴 크기를 지정해 두면 항상 html 요소로부터 글꼴 크기를 상속 받습니다. html 요소의 글꼴 크기를 변경하는 것만으로 페이지의 모든 글꼴 크기를 변경할 수 있다는 점 자체는 em 요소와 다를 바 없지만 rem 요소는 부모 요소로부터 상속을 받지 않기 때문에 페이지 모든 글꼴의 크기 변화를 예측할 수 있고 계산하기도 쉽습니다.

#### • 유용한 상황

단말기의 해상도에 따라 기본 글꼴 크기를 전체적으로 다르게 제어할 필요가 있을 때 미디어쿼리와 함께 rem 단위를 사용하면 효과적입니다.

em 단위의 기준 값 변경 코드를 응용하여 px 단위와 비슷하게 사용하기 위해 html 요소의 폰트 사이즈를 62.5%로 선언하면 아래와 같이 보다 쉽게 사이즈를 계산하여 사용할 수 있습니다.

```
html { font-size:62.5%; } /* ==10px */
body { font-size:1.4rem; } /* == 14px */
h1 { font-size:2.4rem; } /* == 24px */
p { padding:1.4rem; } /* == 14px */
li { margin:1.4rem; } /* == 14px */
```

기본적으로 5px 미만을 제외하고는 모두 rem 단위를 사용하여 작업하도록 합니다. 또한 기본 디자인 사이즈는 기본 디바이스 해상도의 2배 크기로 제작되므로 rem 단위 환산 시 반드시  $\div 2 \times 0.1$ 을 해 주어야 합니다.

#### px → rem단위 환산 계산법

디자인 파일의 이미지 width 값이 100px이라 할 때

$100 \div 2 \times 0.1 = 5\text{rem}$

## 3. CSS3 속성

속성을 선언할 때는 그 쓰임새가 레이아웃과 관련이 큰 것에서 시작한다.  
 심표로 구분되는 선택자 간 공백은 제거한다.

### • 속성 선언 순서

속성을 선언할 때는 그 쓰임새가 레이아웃과 관련이 큰 것에서 시작하여 레이아웃과 무관한 것 순서로 선언합니다. 아래 표와 같이 대표되는 속성 다음으로 의미 순서대로 선언합니다.

- 속성 선언 순서 기준

: 레이아웃 -> Box -> font -> 기타 -> border -> background

순서	의미	대표되는 속성
1	레이아웃	display / visibility / overflow / float / clear / position top, right, bottom, left / z-index
2	BOX	width / height / margin / padding / line-height
3	font	font-size / color / letter-spacing / word-spacing / text-align / text-decoration / text-indent / vertical-align white-space
4	border	border / border-radius /
5	background	background / background-size /
6	기타	위에 언급되지 않은 나머지 속성들로 폰트의 관련 속성 이후에 선언하며, 기타 속성 내의 선언 순서는 무관함 벤더속성, 핵속성

```
.btn{*border:2px solid #f60;-webkit-boder-radius:2px;}
```

### • 공백

- 선택자 간 공백 제거 : 심표로 구분되는 선택자 간 공백은 제거합니다.

잘못된 예	올바른 예
a:hover, ^a:active, ^a:focus{color:#fff;}	a:hover,a:active,a:focus{color:#fff;}

- 속성 간 공백 제거 : 속성 간 공백 및 속성 값 사이 공백은 제거합니다.

잘못된 예	올바른 예
.class p{color:#000; ^line-height:18px;}.class p{color:#000^line-height:18px;}	.class p{color:#000;line-height:18px;}

- 종괄호 좌우 공백 제거 : 종괄호 좌, 우 공백은 제거합니다.

- 마지막 세미콜론 반드시 사용 : 마지막 속성의 세미콜론(;)은 반드시 넣어줍니다.

잘못된 예	올바른 예
.class p^{color:#000;}	.class p{color:#000;}
.class p{^color:#000;line-height:18px;^}	.class p{color:#000;line-height:18px;}
.class{font-size:12px;color:#000}	.class{font-size:12px;color:#000;}

## ● Markup Basic Interaction

1. 서비스 디바이스 별 최적 해상도
2. 기본 선언 및 폴더 구조
3. 생산성 향상을 위한 Tip
4. 앱/웹 접근성 준수



## ● 하이브리드앱 Markup Basic Interaction

### 1. 서비스디바이스 별 최적 해상도 - Iphone3/3G, iphone4/4s 및 Galaxy S2 기준

iOS 와 Android 플랫폼 모두 지원하는 어플리케이션 제작 시 640x960 pixel 해상도 기준으로 제작 (디자인요청)

iPhone 3G/3GS



320 x 480 픽셀

iPhone 4/4s



640 x 960 픽셀

Galaxy S2



480 X 800 픽셀

## ● 하이브리드앱 Markup Basic Interaction

### 1. 서비스디바이스 별 최적 해상도 - iphone5 및 Galaxy S3 기준

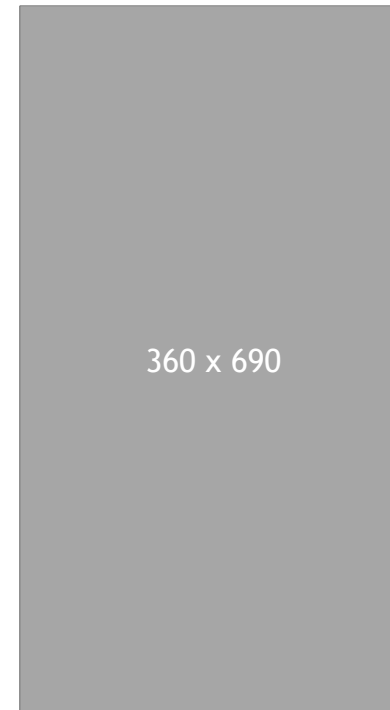
iOS 와 Android 플랫폼 모두 지원하는 어플리케이션 제작 시 640x1136 pixel 해상도 기준으로 제작 (디자인요청)

iPhone 5/5c/5s



640 x 1136 픽셀

Galaxy S3



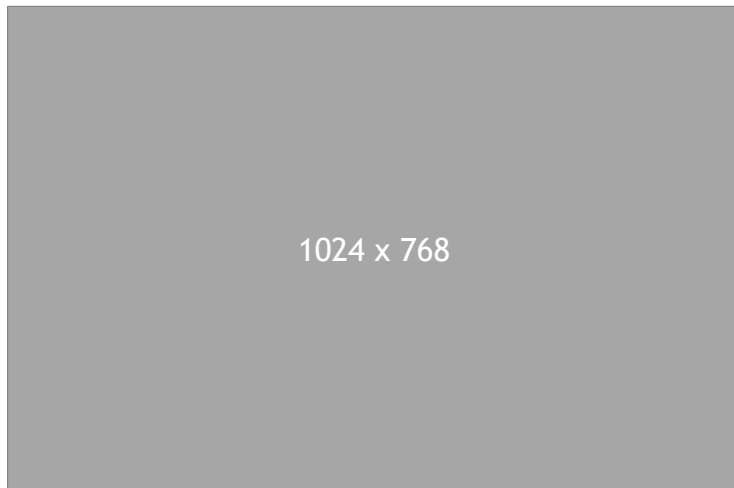
720 X 1280 픽셀

## ● 하이브리드앱 Markup Basic Interaction

### 1. 서비스디바이스 별 최적 해상도 - iPad2 및 Android Tablet 10.1 기준

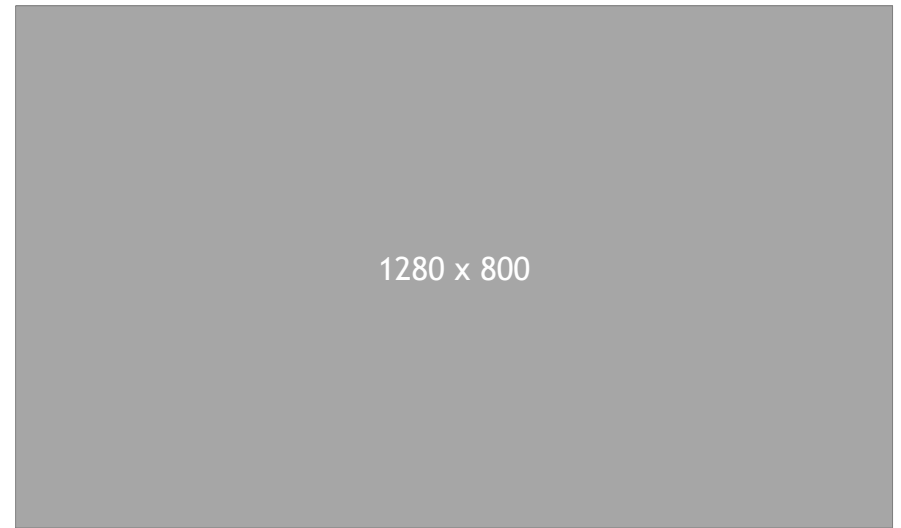
iOS 와 Android 플랫폼 모두 지원하는 어플리케이션 제작 시 1024 x 768 pixel 해상도 기준으로 제작 (디자인요청)

IPad I, II, mini (iOS)



1024 x 768 픽셀

Tab 10.1



1280 x 800 픽셀

## 2. 기본 선언 및 폴더 구조

Phone, Tablet 구분 없이 제작 시 공통으로 지켜야 할 선언과 환경은 다음과 같다.

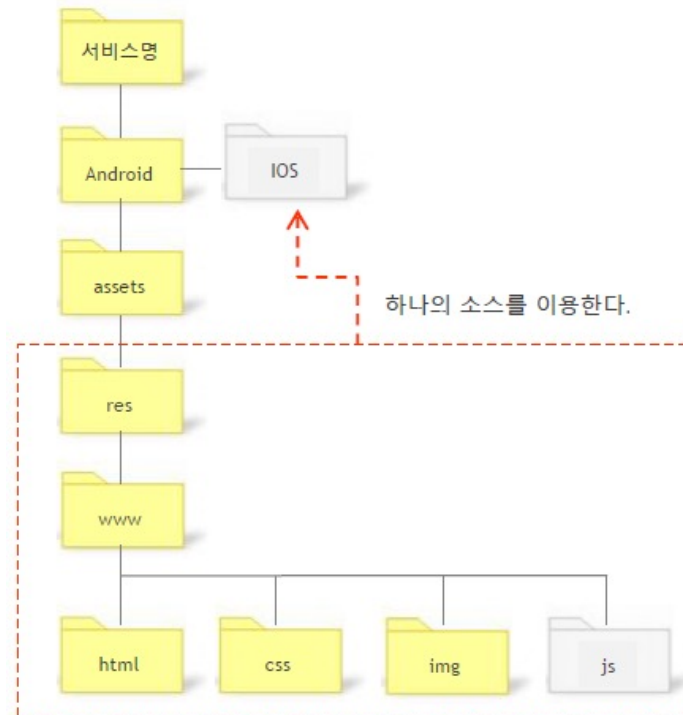
### • 기본선언

구분	환경	
platform	하이브리드앱	
HTML	HTML 5	<!DOCTYPE html>
CSS	CSS3	morpheus_reset.css morpheus_common.css morpheus_style.css
인코딩	utf-8	<meta charset="utf-8">
OS	iOS	iOS 5(4) ~
	Android	Android 4.3(2.3.3) ~
javaScript	표준 JavaScript	
viewport	<meta name="viewport" content="width=device-width, maximum-scale=1.0, minimum-scale=1.0, user-scalable=no, target-densitydpi=medium-dpi" />	

### • 폴더 구조

마크 업을 위한 폴더를 생성할 때 폴더 구조는 다음 그림과 같다.

- 작업 폴더 내 구성 : 이미지 폴더와 CSS 폴더는 기본으로 생성되어 있다. 각 폴더 내 파일의 개수가 많거나, 별도의 폴더로 그룹핑하여 관리하는 것이 편리할 경우, 하위 폴더를 생성할 수 있다.
- 이미지 폴더의 경우 되도록이면 하위폴더를 생성하지 않는 것이 좋다.



### 3. 생산성 향상을 위한 Tip

HTML, CSS로 개발된 화면에 대한 확인은 Chrome브라우저에서 미리 확인해 볼 수 있습니다.

#### • Chrome 개발자 도구(Android)

Device 에서 보이는 것과 100% 동일하지는 않지만 거의 유사하다고 볼 수 있기 때문에 작업 속도 개선을 위해서 Chrome 브라우저에서 우선 확인을 하고, 최종 확인은 Device에서 한 후 세부 수정을 하는 순서로 작업합니다.

Chrome 브라우저에서 모바일 환경에 맞게 화면에 대한 확인을 하려면, 개발자도구(F12)의 설정 화면을 엽니다.

**개발자 도구에서 각 Device별 로도 확인하실 수 있습니다.**

#### • 리모트 디버깅(iOS6이상)

Chrome브라우저의 개발자도구를 통해서 디버깅을 하듯이, iOS환경에서도 Web View화면을 Safari브라우저로 디버깅 할 수 있습니다.

1. iOS Simulator에서 설정 > Safari > 고급으로 들어가면 다음과 같은 화면에서 Web Inspector항목을 ON으로 활성화합니다.
2. 컴퓨터에서 Safari브라우저의 환경설정에서 '메뉴막대에서 개발자용 메뉴보기'를 체크합니다.
3. Safari브라우저의 메뉴막대에서, 개발자용 > iPhone Simulator 메뉴를 통해서 iPhone Simulator에 표시된 페이지에 대해서 디버깅을 합니다.

#### 4. 웹표준 준수

### HTML 표준 Check (HTML Validation Check)

#### • Validation Check

##### 1. W3C Validation Check 사이트 접근을 통한 유효성 점검

<http://validator.w3.org/>

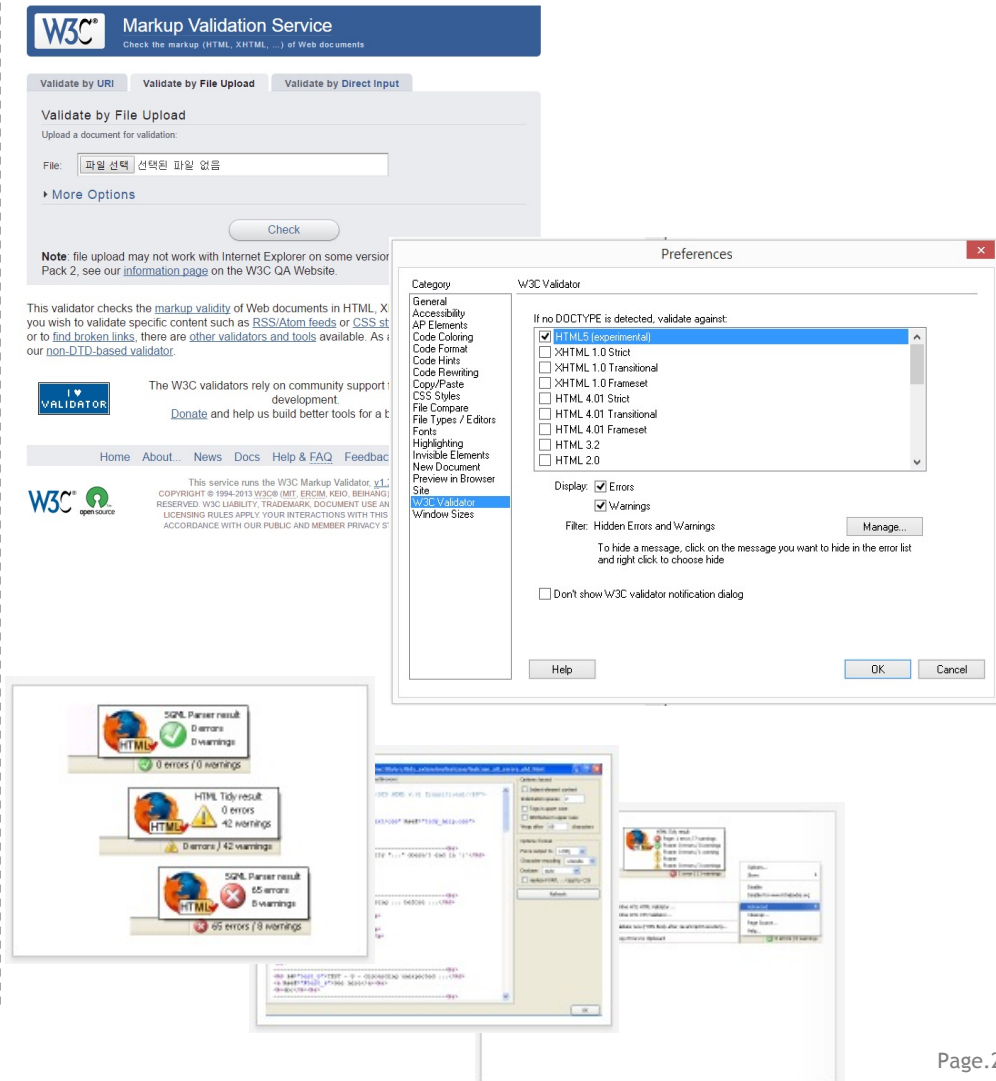
<http://validator.kldp.org/>

##### 2. 보안된 페이지의 HTML Validation Check

: 신뢰할 수 있는 도구를 이용한 유효성 점검

: [Dreamweaver > File > Check Page > Validate Markup\(Shift+F6\)](#)

: [Firefox](#) + [HTML Validator](#)



## 4. 접근성 준수

모바일 애플리케이션 접근성 지침(행안부고시2011-38호) 에 준수하여 마크 업 하여야 한다.

### • 모바일 애플리케이션 접근성 지침이란

모바일 애플리케이션 서비스 제공자가 장애인과 고령자 등의 접근성\*을 보장하기 위해 애플리케이션 제작 시 지켜야 할 사항을 규정한 내용입니다.

### • 적용범위

OS 를 갖는 모바일 전화기, 태블릿 기기, 전자책 기기에 적용

### • 모바일 애플리케이션 접근성 지침 > 준수사항(7개)

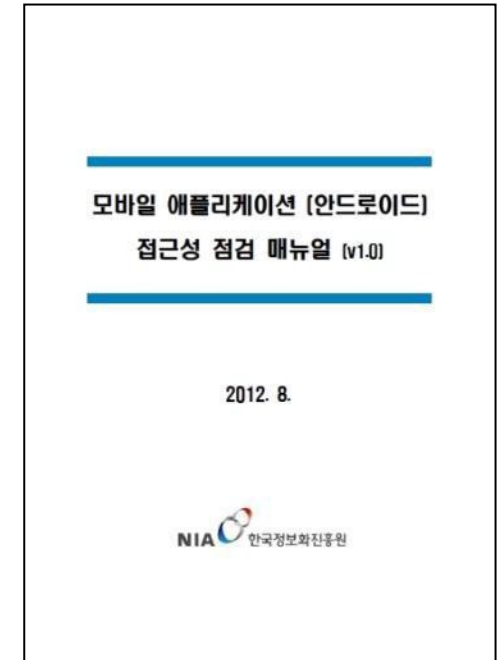
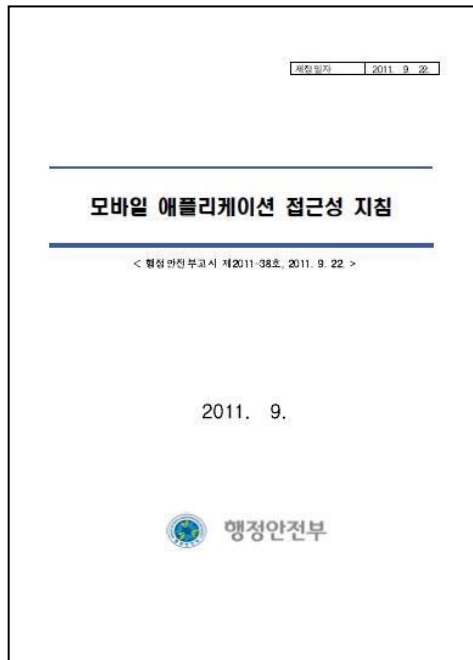
No	지침	내용
1	대체텍스트	• 비 텍스트에 콘텐츠 의미나 정보를 동등하게 인식할 수 있도록 대체 텍스트를 짧고 명확하게 제공
2	초점	• 모든 객체에 초점(Focus)를 제공, 순차적 이동될 수 있도록 제공 • 선택된 초점은 테두리나 하이라이트로 표시, 논리적 순서로 제공
3	OS 접근성 기능 지원	• 각 모바일 OS에서 장애인을 위해 제공하는 기능과 호환될 수 있도록 서비스 제공
4	누르기 동작 지원	• Multi-Touch, Slide, Drag & Drop 등 복잡한 동작을 One Tap(단순동작)으로 대체 가능해야 함
5	색에 무관한 인식	• 색각 이상자도 정보에 동등하게 접근 하도록 무늬, 패턴 제공 - 정보는 색과 무관하게 인식 가능. - 색으로 정보 구분한 경우 타 방법으로도 동일 내용 전달 가능. - 흑백화면에서도 동일 내용 전달 가능.
6	명도대비	• 저 시력자, 고령자 등을 위해 전경과 배경을 구분할 수 있도록 고대비 제공 -전경색과 배경색의 명확한 구분 -고대비 제공이 불가능 할 경우 설정에서 명도 조절 기능 제공
7	자막, 수화 제공	• 동영상에 대한 내용을 동등하게 인식할 수 있도록 자막, 원고 또는 수화를 제공

### • 모바일 애플리케이션 접근성 지침 > 권고사항(8개)

No	권고	내용
1	기본 사용자 인터페이스 컴포넌트	• Native UI Component(대화상자, 버튼, 체크박스, 타이틀 바 등)로 최대한 제공 보조기기와의 접근성 확보를 위해 적극 활용 권장
2	컨트롤 간 충분한 간격	• 사용자 의도와 다른 컨트롤을 누르지 않도록 터치 영역 포함한 컨트롤 간 충분한 간격 제공 • 컨트롤 중심간 간격은 13mm 이상 권장
3	알림 기능	• 진동, 시각, 소리 등 다양한 방법으로 사용자에게 알림 및 선택 기능 제공 • Native UI를 활용하는 것이 바람직함
4	범용폰트 사용	• Global Font로 제공하여 확대, 축소, 기울임 등 변형 가능하도록 제공 • 어플리케이션 전 화면에서 폰트 크기 조절 가능하도록 설계, 최소한 확대 제공 • 이미지 텍스트 지양
5	사용자 인터페이스 일관성	• 학습 필요 없도록 UI요소를 일관성 있게 배치 • 버전이 바뀌어도 배치는 일관성 유지
6	깜박거림의 사용제한	• 광과민성 발작을 일으킬 수 있는 콘텐츠는 제공하지 않음 • 깜박이거나 번쩍이는 객체 사용을 지양 • 꼭 제공할 경우 초당 3~50회 주기는 피해 설계
7	배경을 사용 금지	• 자동 재생 배경음(동영상, 음악, 음성 안내 등) 사용 지양 • 사용할 경우, 사용자가 쉽게 멈춤, 정지, 음량조절을 제어할 수 있게 제공
8	장애인 등 사용자 평가	• 장애인의 이용보장을 위해 장애인 대상의 사용자 평가 수행 • 다양한 모바일 기기에서 이용가능 여부 점검 • 어플리케이션 출시 전 장애인, 고령자 등 대상으로 Usability Test(사용성 테스트)실시 • Usability Test 대상은 시각, 청각, 뇌병변, 지적, 지체 장애 및 고령자 • 구체적 결과를 별도 공시 하는 것이 바람직 함

### 4. 접근성 준수

- 모바일 애플리케이션 접근성 지침(행안부고시제2013-107호,2013년8월)  
[http://wah.or.kr/board/boardView.asp?page=1&brd\\_sn=4&brd\\_idx=979](http://wah.or.kr/board/boardView.asp?page=1&brd_sn=4&brd_idx=979)
- 모바일 애플리케이션(iOS) 접근성 점검 매뉴얼1.0(2011년10월)  
[http://wah.or.kr/board/boardView.asp?page=1&brd\\_sn=4&brd\\_idx=719](http://wah.or.kr/board/boardView.asp?page=1&brd_sn=4&brd_idx=719)
- 모바일 애플리케이션(안드로이드) 접근성 점검 매뉴얼1.0(2012년8월)  
[http://wah.or.kr/board/boardView.asp?page=1&brd\\_sn=4&brd\\_idx=806](http://wah.or.kr/board/boardView.asp?page=1&brd_sn=4&brd_idx=806)





END of Document

Thank you!



Uracle UX Team