

Mobile Web

대개 스마트폰에서 검색하는 웹 페이지가 바로 일반적인 모바일 웹입니다.

모바일 웹은 모바일에서도 PC용 사이트의 글자 폰트와 이미지, 터치아이콘등 데스크탑 브라우저에서 실행되는 기능을 모바일에 맞추어 표현한 사이트로서 PC용 홈페이지를 모바일 스크린 크기로 줄여 놓은 것이라고 할 수 있습니다.

이러한 이유로 대개 모바일 웹은 일반 PC용 웹 페이지에 비해서 간단한 UI(User Interface)를 가지고 있습니다.

모바일웹은 웹 기술로 개발되어 모바일 브라우저에서 실행되는데 풀 브라우징 방식으로 페이지를 이동해 앱에 비해 접속 속도가 느리다는 단점을 가지고 있습니다.

Full Browsing(풀 브라우징)

휴대폰 인터넷 콘텐츠를 PC화면에서 보는 것과 같은 동일한 인터넷 화면으로 제공하는 서비스로서 각종 문서, 이미지, 동영상 등을 볼수 있도록 도와준다.

또한 직접 URL 입력을 통해 유선 웹포털로 직접 접속이 가능해 이동통신사 무선 포털이 제공하는 콘텐츠의 범위 이상을 제공한다.

모바일 웹에서 사용되는 기술에는 다음과 같은 것들이 있습니다.

- HTML5
- CSS3
- 자바스크립트
- jQuery Mobile, Sencha Touch
- PhoneGap, Titanium, Ionic, Rramework7, OnsenUI, 구글의 Material Design

모바일 환경에 최적화된 웹 브라우저들에는 IOS 에 탑재된 모바일 사파리, 안드로이드의 기본 브라우저가 대표적입니다.

이 외에 각 모바일 플랫폼에 호환되는 오페라 미니 브라우저, 파이어 폭스 모바일 브라우저 등의 다양한 웹 브라우저를 선택하여 사용할 수 있습니다.

기기별 다양한 해상도

모바일 웹을 위한 HTML 마크업하기 위해서 가장 중요한 것은 해상도입니다.

일반적으로 PC 기반의 웹 페이지는 1024*768 , 1280*1024 등과 같이 고정된 해상도에 최적화하도록 사이트 디자인 및 마크업을 진행했습니다.

하지만 모바일의 경우는 320*480 , 640*960 , 480*800 , 1024*600 등 모바일 기기마다 해상도가 다르기 때문에 모바일 웹을 개발할 때는 더 다양한 해상도를 고려해야 합니다.

그리고 해상도는 달라도 액정크기는 대부분 비슷합니다.

즉, 이 말은 같은 크기로 제작할 경우 액정 크기와 해상도가 높은 모바일 기기에서는 상대적으로 웹 페이지가 작게 보인다는 말이 됩니다.

그러므로 모바일 웹을 제작할 때는 가능한 구글 크롬 Webkit 기반의 웹 브라우저에서 테스트를 하는 것이 좋습니다. 또 모바일에서는 픽셀(px) 단위의 딱 맞는 디자인이 아닌 퍼센트(%) 단위로 웹 페이지를 제작하는 것이 좋습니다.

다양한 해상도에 대응하는 방법은 크게 두가지입니다.

하나는 CSS3 미디어쿼리(media query)를 이용해 크기에 사이트를 맞추는 방법이고 다른 하나는 모바일 기기의 폭에 사이트를 맞추는 방법입니다.

모바일 기기의 폭에 사이트를 맞추는 방법은 HTML의 <meta> 태그에 viewport 속성을 설정하는 것입니다.

Viewport(뷰포트)

뷰포트라는 개념자체는 모바일 웹에 국한된 것은 아닙니다.

뷰포트를 대략 정의하자면, 컴퓨터나 휴대 단말기 등 장치에 Display 요소가 표현되는 영역을 말합니다.

모바일 웹의 경우 아이폰 사파리 브라우저에서 웹 페이지가 표시되는 영역으로 해석할 수 있습니다.

그렇다면 왜 모바일 웹에서는 뷰포트가 중요해 진 것일까?!

이는 바로 모바일 웹이 웹 페이지를 브라우징하는 특징에 기인하고 있습니다.

현재 스마트폰 브라우저는 풀브라우징을 지원하고 있습니다.

풀브라우징은 모바일 환경에서도 데스크탑 환경처럼 웹화면 전체를 자연스럽게 브라우징 할 수 있는 환경을 말하는데

작은 화면의 모바일 단말기에 웹 페이지 모두 표시하려다 보니 전체적인 페이지 배율(비율) 조정이 가해지게 됩니다.

즉, 화면에 맞도록 전체적으로 축소되어 화면 모두가 브라우징 되지만 페이지에 있는 콘텐츠들의 가독성은 상당히 떨어지게 되는 것입니다.

웹킷 기반 브라우저의 뷰포트 기본 가로 너비는 980 픽셀입니다. 하지만 아이폰 사파리브라우저의 가로 너비는 320 픽셀입니다.

이 의미는 980 크기의 페이지를 320 화면에 다 보이도록 전체적인 배율(비율)축소가 발생된다는 뜻입니다.

뷰포트 메타태그를 이용하면 모바일 브라우저의 뷰포트 크기나 배율등을 조정할 수 있어 모바일 웹에 최적의 상태로 화면이 표시되도록 설정할 수 있게 됩니다.

모바일 브라우저에 대응하는 html 문서의 head 안에 veiwport 를 설정합니다.

뷰포트는 브라우저와 해상도에 따라 다르게 설정할 수 있습니다.다.

뷰포트의 설정이 반드시 필요한 이유는 뷰포트 설정에 따라서 확대나 축소와 같은 화면의 배율(비율)을 조정하고, 모든 HTML이 사용해야 하는 기본 크기를 **device-width** 와 같이 기기 크기에 맞추기 위해서 입니다.

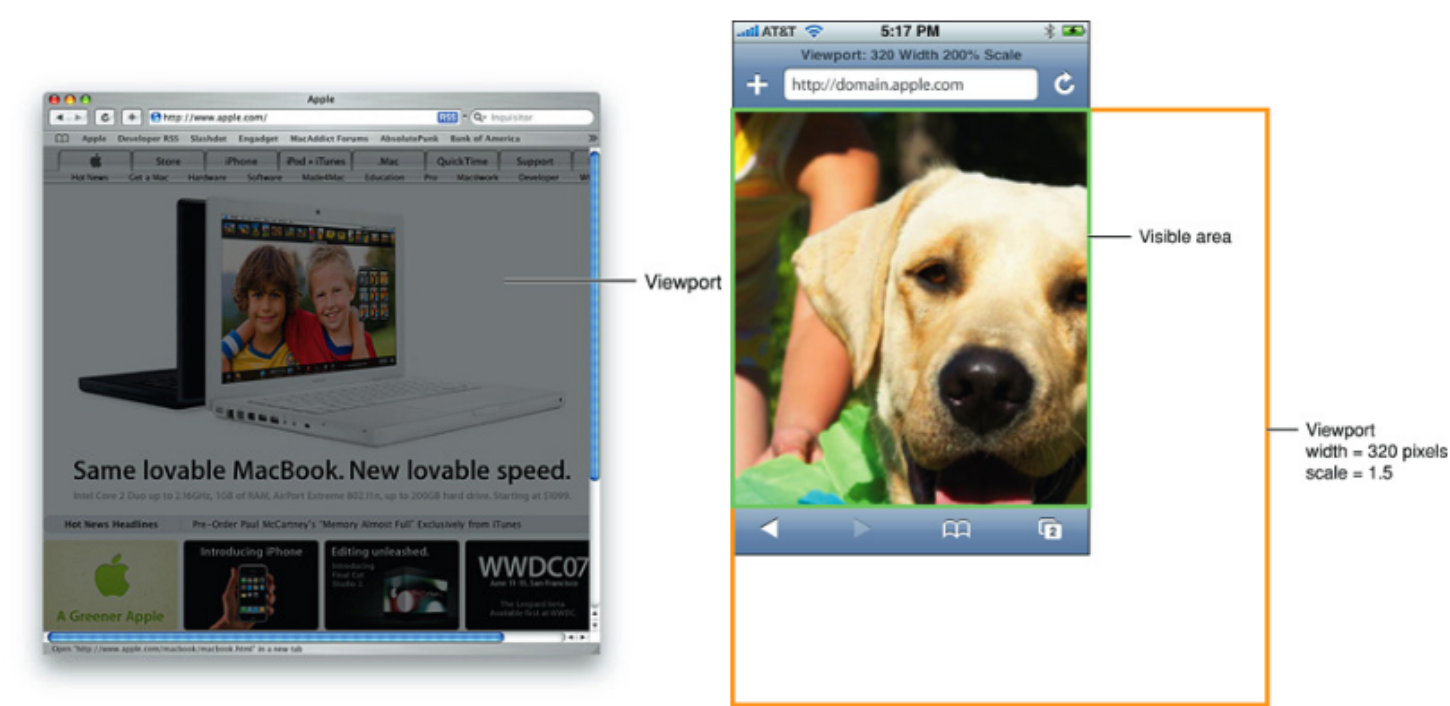
사실 뷰포트 설정은 데스크톱에서는 전혀 영향이 없습니다.

이는 데스크톱에 설치된 브라우저들의 경우 뷰포트의 기본값을 사용하는 것이 아니라 모니터 해상도를 기준으로 사용하기 때문에 전혀 영향을 주지 못합니다.

그래서 다양한 상황을 적극적으로 대처하려면 기본적으로 뷰포트를 설정하고 CSS3 미디어 쿼리를 사용하는 것입니다.






데스크톱/모바일 뷰포트

데스크톱은 화면에 보이는 영역 이 뷰 포트가 되지만 모바일은 문서 크기 가 뷰 포트가 됩니다.



뷰 포트 의 물리적인 너비가 320px 이면 웹 문서 의 320px이 뷰 포트에 들어가게 됩니다.

iPhone4+ 레티나 디스플레이 는 640px 너비의 뷰 포트를 지녔지만 웹 문서의 320px만 들어간다.(웹 문서의 1px을 레티나 장치에서는 4px로 표시하기 때문)

					
	3.5"	3.5"	4"	4.7"	5.5"
아이폰 해상도	iPhone 3, 3GS	iPhone 4, 4S	iPhone 5, 5S	iPhone 6	iPhone 6 plus
비율	3:2	3:2	16:9	16:9	16:9
PPI	163 ppi	326 ppi	326 ppi	326 ppi	401 ppi
브라우저 해상도	320 x 480	320 x 480	320 x 568	375 x 667	414 x 736
이미지 해상도	320 x 480	640 x 960	640 x 1136	750 x 1334	1242 x 2208
디바이스 해상도	320 x 480	640 x 960	640 x 1136	750 x 1334	1080 x 1920

meta 태그의 viewport 속성

```
HTML

<meta name="viewport" content="width=device-width, initial-scale=1, ...">

width
```

- 모바일 기기의 폭을 설정
- device-width로 값을 설정하면 모바일 기기의 폭에 맞춰서 보이게 할수 있다.
- 단위로는 px을 이용, 기본값은 980px, 최소값은 200px이고 최대 10,000까지 지정가능하다.
- device-width 로 설정하면 기기의 화면 크기를 기준으로 삼는다.

initial-sacle

- 페이지가 보이는 기본 비율(화면의 배율을 의미)
- 1 이면 모바일 기기의 가로 해상도로 맞춰진다.
- 0 - 10 까지의 배율을 조정할 수 있다.

maximum-scale

- 페이지가 보이는 최대 비율.
- 즉, 사용자가 줌 인을 할 때의 확대하는 비율

minimum-scale

- 페이지가 보이는 최소 비율.
- 즉, 사용자가 줌 아웃을 할 때의 줄어드는 배율을 의미. 기본값은 0.25씩

user-scalable

- 페이지의 확대/축소 가능 여부를 정의.

타 사이트 뷰포트 참고

국내 포털 및 타사이트에서 모바일 웹 사이트를 위해 설정한 뷰포트 태그를 참고해 보도록 합니다.

m.naver.com

HTML

```
<meta name="viewport" content="width=device-width,initial-scale=1.0,minimum-scale=1.0,maximum-scale=1.0">
```

m.daum.net

HTML

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

m.nate.com

HTML

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

m.yahoo.com

HTML

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

위 주요 사이트들의 뷰포트는 거의 동일하며 일반적으로 이 구성으로 뷰포트를 많이 설정하고 있습니다.
위 뷰포트는 가로 너비는 장치의 너비를 지정하였으며 사용자에게 의해 확대/축소가 되지 않도록 설정하고 초기 스케일은 1.0 즉 확대/축소 없는, 본 크기 그대로를 지정한 것입니다.

모바일 웹 제작 기초

HTML 마크업 및 CSS 를 이해하고 있다는 가정하에 모바일 웹을 만들기 위한 기본적인 방법에 대해서 알아보도록 하겠습니다.

모바일 웹을 위한 HTML 마크업 방법

결론부터 이야기하자면 모바일 웹의 HTML 표준 마크업을 함에 있어서 특별한 방법이 존재하지는 않습니다.
기존에 웹 표준에 기반을 둔 HTML 마크업을 하는 방식과 크게 다르지 않기 때문입니다.
단, 모바일 웹을 위한 UI/UX 그리고 모바일 웹에 최적화되어 있는 태그는 고려해야 합니다.

모바일 웹을 위한 텍스트 크기 설정

모바일에서 폰트 단위는 pt, in 등 절대적인 크기를 나타내는 속성값보다 px, em, ex, % 등의 상위 요소의 크기에 영향을 받는 상대적인 크기를 나타내는 속성을 사용해야 합니다.
다양한 해상도와 기기의 액정 크기를 생각하면 텍스트 크기를 절대적인 크기보다는 유동적인 상대적 크기로 제작하는 것인 접근성, 사용성 면에서 더욱 좋습니다.

일반적으로 많은 사람들이 **em** 단위를 쓰면 해상도에 따라서 크기가 달라진다고 오해하는 경우가 종종 있습니다.

하지만 em 을 쓰는 이유는 IE6과 같이 **zoom** 기능이 없는 브라우저에서 브라우저의 글꼴 크기 옵션을 조절했을 때 상대적으로 크거나 작게 동작하게 하기 위해서 입니다.

IE7+, 모던 브라우저는 모두 zoom 기능을 지원하고 있는 덕분에, 사실 **em** 대신 **px** 을 사용해도 무방합니다.

사실 em 으로 취할 수 있는 이득이 거의 없어졌지만, Responsive Web Design 을 추구하는 작업을 하거나 화면 크기에 따라서 동적으로 **body의 font-size** 를 조절하는 등의 특수한 작업을 하실 경우에는 em 을 활용하여 scaleable 한 UI 를 구성할 수도 있습니다.

또한 해외에서는 유동성 부분 때문에 대부분이 em 을 사용하며, 장래에는 rem 이 표준으로 잡아갈 것으로 보입니다.

상대단위 (px, em, ex, %, rem)

- **px** : pixel을 기준으로 브라우저에는 고정적이지만 표시장치(기기의 해상도)에 따라서 상대적인 크기를 갖는다. (1px = 1dot)
- **em** : 해당폰트의 대문자 M의 너비를 기준으로 값을 가진다.
- **ex** : x-height. 해당 폰트의 소문자 x의 높이를 기준으로 값을 가진다.
- **%** : percent. 기본글꼴의 크기에 대해 상대적으로 값을 가진다.
- **rem** : root em. 최상위 요소의 크기에 대해 상대적으로 값을 가진다.

브라우저의 pt 단위와 포토샵의 pt 단위는 다르며, 포토샵에서 텍스트를 pt 단위로 사용 시 해당 폰트의 높이가 px 단위가 된다.

권장 단위

상대적인 단위인 em을 사용시 보다 유동적인 환경으로 접근성을 구현할 수 있지만, 브라우저 간의 디자인 요소(간격의 오차)를 고려하는 경우에는 제약이 많다.
이에 다른 단위보다는 PC에서는 브라우저 호환성을 위해 px 단위의 사용을 권장한다.

단, 모바일과 같이 유동적인 레이아웃을 구현하기 위해서는 상대단위(em,%)를 권장하고 있는데,
아이폰4(iOS 4)의 보이스오버 기능과 관련하여 다음과 같은 IR 기법 이슈가 있으니 주의 하여야 한다. (iOS 5 제외)

- iOS 4에서는 IR(park method) 기법 사용 시 text-indent의 단위에 따라 보이스오버 기능과 관련된 퍼포먼스 이슈가 발생한다.
- iOS 4에서 em 단위는 px 단위에 비해 6배~10배까지 성능이 하락하며, in 단위는 15배~24배까지 성능이 하락한다. (px 단위의 지연시간을 0.1이라고 설정했을 때)

폰트사이즈 계산 방법

브라우저의 기본 값은 1em = 16px 이므로 12px 을 사용하고자 할 때, em 계산법은 12(사용하고 싶은 px) / 16(기준크기 px) = 0.75em 이 된다.

% 로 변환할 시, 12(사용하고자 하는 px) / 16 (기준크기 px) * 100 = 75% 가 된다.

모바일 웹 바로가기 아이콘

스마트폰이나 태블릿에서는 기본적으로 모바일 웹 페이지를 즐겨찾기 하여 홈 화면에 **앱 아이콘** 형태로 배치할 수 있습니다.

원하는 웹 페이지를 바로 볼 수 있도록 편의성을 고려한 것입니다.

기존 PC 기반의 웹에서는 **파비콘**이라는 바로가기(즐겨찾기) 아이콘을 제작하듯이 스마트폰이나 태블릿에서는 아이콘의 해상도가 좀 더 높기 때문에 따로 제작하는 경우가 많습니다.

다음은 PC 웹에서 사용되는 파비콘 설정에 대한 것입니다.

HTML

```
<!doctype html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>PC 웹 파비콘 설정</title>
  <link rel="shortcut icon" type="image/x-icon" href="http://webclub.tistory.com/images/favicon.ico" />
</head>
<body>

</body>
</html>
```

파비콘 사이즈는 16*16 으로 제작합니다.

rel 속성은 a 요소, link 요소, area 요소에서 사용되며, 링크된 대상과의 관계를 지정합니다.
area 요소는 하이퍼링크, 링크에 대응하는 이미지맵 상의 영역, 이미지맵 내의 비활성 영역을 나타냅니다.

아이콘 파일은 PNG 형식으로 저장하며 크기는 다음과 설정할 수 있습니다.

HTML

```
<!doctype html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>모바일 웹 및 앱 아이콘 설정</title>

  <!-- 아이폰 전용 태그 -->
  <link rel="apple-touch-icon" href="app_icon.png">

  <!-- 안드로이드, 아이폰 모두 사용 가능한 태그 -->
  <link rel="apple-touch-icon-precomposed" href="app_icon.png">

</head>
<body>

</body>
</html>
```

아이콘 크기는 디바이스에 따른 크기별로 만드는 것은 불편하고 낭비이므로 iOS7+ 이상을 고려하여 120*120 사이즈로 제작하는 것이 좋습니다.
일반적으로 아이콘 이미지에 대한 사이즈 및 제작은 디자이너의 몫이기 때문에 자세히 알 필요는 없을 것 같습니다.

모바일 웹을 위한 텍스트 크기 설정

모바일 환경은 작은 화면에서 제공되므로 너무 작은 텍스트는 피하고 적절한 크기로 보여야 합니다.
그래서 가로 모드와 세로 모드로 변화하는 모바일 환경에서는 폰트가 너무 커지거나 너무 작아지면 사용자들이 불편하기 때문에 이에 대한 대응으로 텍스트 확대 방지 태그가 나왔 습니다.
예를 들어 모바일 사파리 브라우저는 텍스트 영역의 가로를 결정하는 속성값에 맞춰 텍스트 텍스트 크기를 자동으로 조절하거나 크기를 고정시킬 수 있습니다.
즉, 아이폰의 경우 뷰포트가 변경이 되면 자동으로 폰트 크기를 조절합니다.
이것을 방지하는것이 text-size-adjus 입니다.
이를 위해서 사용하는 속성은 -webkit-text-size-adjust 입니다.

CSS

```
html {
  -webkit-text-size-adjust: none;
}
```

- auto : 화면의 폭에 맞게 텍스트의 크기를 자동으로 조절한다.
자동으로 조절한다는것은 디바이스의 렌더링이 다르다면 다르게 렌더링이 되므로 추천하는 방식은 아니다.

- `none` : 사용자들이 불편함을 느끼지 않도록 폰트의 크기를 자동으로 변환한다.
텍스트의 크기를 자동으로 조절하지 말라는 의미로 일반적으로 많이 사용하는 방법이다.
- `%` : 폰트크기를 명시적으로 지정해주는 것이다.
이 방법도 경우에 따라 자주 사용하는 방법이다.
명시적으로 100%를 주어 모든 디바이스에서 동일하게 보여주게 하는 방식이다.