

6장. 반응형 이미지

목차

[유동형 이미지](#)

[유동형 비디오](#)

[Fitvids.js](#)

[기타자료](#)

[반응형 이미지](#)

[모바일 퍼스트 \(Mobile-First\)](#)

[모바일 퍼스트 접근방법 1](#)

[모바일 퍼스트 접근방법 2](#)

[선택적 이미지 전송](#)

[브라우저 쿠키를 이용한 선택적 이미지 화일 제공 기법](#)

[외부 서비스](#)

[이미지 다운로드 실험](#)

[새로운 반응형 이미지 \(표준\)](#)

[반응형 이미지 그룹 \(<http://responsiveimages.org/> \)](#)

[구현](#)

[picturefill.js](#)

[jquery-picture.js](#)

[기타 반응형 비 텍스트 요소들](#)

[반응형 맵](#)

[반응형 광고](#)

[반응형 슬라이더](#)

[참고문헌/자료](#)

문서정보

권리

이 문서는, sir.co.kr 에서 2012년 11월 2일 부터 진행된 '[반응형 웹](#)' 강좌에 사용하기 위해서 정리한 문서로, 별도의 출처를 밝힌 자료를 제외한 모든 내용의 권리는, 작성자 (전진, jinjunkr@gmail.com) 에게 있습니다. 아직 문서나 문서의 일부의 재사용에 대한 정책을 정하지는 않았지만, 상용화에는 기본적으로 반대합니다.

이 문서에서 사용된 출처를 밝힌 자료/문서/링크/이미지 등은 모두 해당 출처에 그 저작권이 있으며, 그 권리를 침해하려는 어떠한 의도도 없습니다.

이력

- 11/25/2012: 최초 작성
- 05/02/2013: 내용 추가 - HiSRC, art direction by CSS

유동형 이미지

반응형 웹을 제안한 [Ethan Marcotte](#) (이하 EM)가, [유동형 그리드 \(fluid grids\)](#)글과 [함께 2009년에 제안했던 기법](#)으로, 화면의 폭보다 큰 이미지를 화면폭에 맞춰주도록 `max-width:100%`로 주는 것을 뜻합니다.

```
img {  
    max-width: 100%;  
}
```

뿐만아니라, 대부분의 비디오 태그에도 `max-width` 를 적용하는 것으로 대부분의 문제는 해결됩니다.

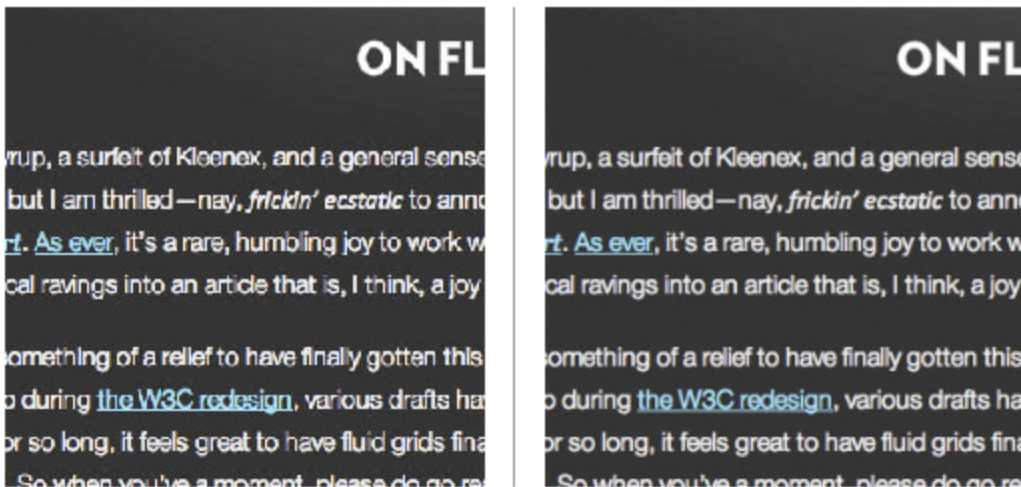
```
img,  
embed,  
object,  
video {  
    max-width: 100%;  
}
```

물론 기대를 저버리지 않는 IE 구버전은 `max-width`를 지원하지 않습니다. 한가지 방법은, 해당 브라우저에서는 `width` 자체를 100%으로 만드는 방법도 있습니다.

```
img {  
    width: 100%;  
}
```

하지만 더 큰 문제는, 큰 이미지가 축소될때 몇몇 OS/브라우저가 제대로 작업을 하지 못하는 경우입니다. (IE 6 과 Windows FF 3 이전버전)

놀랍게도, IE 구버전의 경우, `AlphaImageLoader` 를 사용하면 이 문제가 해결된다고 합니다. IE7에서 [이 방법을 사용하지 않았을때의 이미지](#)와 [사용했을때의 이미지](#)를 보면 차이를 알 수 있습니다. EM이 사용한 스크립트는 [여기서 다운로드](#) 할 수 있습니다.



출처: Fluid Images by EM - <http://www.alistapart.com/articles/fluid-images>

Chris Coyier의 ‘[유연한 이미지 \(flexible images\)](#)’는 max-width와 더불어, height를 auto로 잡아주어서 가로 세로 비율을 ‘강제로’ 유지하는 것도 필요하다고 합니다. (출처: [링크](#)) 또한 -ms-interpolation-mode 를 이용해서 위와 같은 문제를 해결할 수도 있다고 합니다.

```
img {
    max-width: 100%;

    /* just in case, to force correct aspect ratio */
    height: auto !important;
}
img {
    -ms-interpolation-mode: bicubic;
}
```

하지만, [EM의 댓글](#)에서 보면, 위 -ms-interpolation-mode: bicubic이 작동은 하지만, 너무 느리다고 지적하고 있습니다.

유동형 비디오

위 유동형 이미지 글에서 다뤘듯이, 비디오도 간단히 max-width:100% 로 처리한다고 했지만, 몇몇 경우에는 간단히 처리되지 않을 수 있습니다. 구체적인 문제점으로 가로/세로 비율이 맞지 않거나 비디오 오브젝트의 종류에 따라서 달리 처리해야 합니다.

Fitvids.js

Css-tricks.com의 Chris Coyier와 Paravel 이 제작한 [Fitvids.js](#) 는 유동형/반응형 비디오를

처리하기 위한 간단한 jQuery 플러그인 입니다. 사용방법은 다음과 같습니다. (출처: [github/fitvids.js](https://github.com/fitvids.js))

```
<script src="path/to/jquery.min.js"></script>
<script src="path/to/jquery.fitvids.js"></script>
<script>
  $(document).ready(function(){
    // Target your .container, .wrapper, .post, etc.
    $("#thing-with-videos").fitVids();
  });
</script>
```

이러한 코드가 적용되면, 해당 요소 (이 경우는 #thing-with-videos)를 둘러싸는 div.fluid-width-video-wrapper 가 생성되고 필요한 CSS가 적용됩니다. 현재 [처리되는 비디오 소스](#)로는, youtube, vimeo, blip.tv, viddler, kickstarter 라고 합니다.

기타자료

- [Creating Intrinsic Ratios for Video](#): 비디오의 가로세로 비율을 유지하기 위한 기법들
- [CSS: Elastic Videos](#)
- [Create fluid width videos](#): Chris Coyier 글
- [Run-down of handling flexible media](#): 유연한 이미지 이외에 비디오/오디오 등에 대한 설명 (Chris Coyier)
- [This Is Responsive - Resources - Images](#): Brad Frost의 This Is Responsive: Resources 중 이미지 부분

반응형 이미지

그렇다면 유동형 이미지와 반응형 이미지의 차이점은 무엇일까요? 즉 유동형 이미지만으로 해결되지 않는 문제점은 무엇일까요? 대표적인 문제점으로, 데스크탑에서 사용하는 화일크기가 큰 이미지를, 모바일에서 단순히 이미지 가로/세로만 줄인다는 것입니다. 이 문제점을 해결하기 위한 합리적인 방법은, 데스크탑용과 모바일용 이미지를 선별적으로 제공/사용하는 것일것입니다. 반응형 이미지는, 이러한 선별적 이미지 제공/사용을 처리하기 위한 기법들을 뜻합니다. (참고로, 본 강좌에서는, ‘반응형 이미지’가 ‘유동형 이미지’를 포함하는 개념/기법으로 사용합니다.)

다음은 [이전에 포스팅한 제 블로그 글](#)을 발췌했습니다.

반응형 이미지의 선택적 화일 사용은 기본적으로 데스크탑용 이미지 화일과 모바일 기기용 (여러크기의) 이미지 화일들을 미리 준비해두어야 합니다. 이렇게 준비된 기기별 이미지를 전송하는 접근 방법은, 모바일용 이미지를 기본으로 보내는 방법 (모바일 퍼스트)과 기기에 따라서 선별적으로 보내는 방법으로 나눌 수 있습니다.

모바일 퍼스트 (Mobile-First)

기본적으로 모바일용 작은 이미지는 기본적으로 로딩을 하고, 데스크탑 환경에서는 큰 이미지 화일을 별도로 '더' 다운 받는 방법입니다.

모바일 퍼스트 접근방법 1

출처: filament group - [Responsive Images Experimenting with Context Aware Image Sizing](#)

.htaccess와 별도의 js ([responsiveimgs.js](#)) 를 이용하여, 화면폭이 480px보다 클때 큰 이미지 화일을 다운받는 방법입니다. 사용방법은 다음과 같습니다.

```
<script src="responsiveimgs.js"></script>

```

이를 사용하기 위해서는 mod_rewrite가 허용된 서버에서 다음과 같은 [.htaccess](#)를 설정해야 합니다.

```
# Responsive Images
# Mobile-First images that scale responsively and responsibly
# Copyright 2010, Scott Jehl, Filament Group, Inc
# Dual licensed under the MIT or GPL Version 2 licenses.
# //Start Responsive Images
RewriteEngine On
# direct image requests to temp
RewriteCond %{QUERY_STRING} full=(.*)&?
RewriteRule (.*)rwd-router/.*\.(jpe?g|png|gif|webp) $1$1 [L]
# ignore trap for non-image requests, rewrite URL without trap segment
RewriteRule (.*)rwd-router/(.*)$ $1$2
# //End Responsive Images
```

단점은, IE8 미만에서는 작동하지 않고 기본 (작은) 그림만 보여주는 것이고, 스크린 경계치를 하나 (480px 기본이고, [옵션을 주어서 다른 경계치를 사용](#)할 수 있습니다.).

모바일 퍼스트 접근방법 2

출처: [Responsive Images Using CSS3](#)

방법1과는 좀 다른 접근 방법으로, css3의 media-query와 data 필드, attr()함수를 이용하여 스크린 크기에 따라 선택적으로 큰 이미지 화일로 교체하는 방법입니다.

```

```

```
@media (min-device-width:600px) {
  img[data-src-600px] {
    content: attr(data-src-600px, url);
  }
}
```

위 1번 방법과 비교하면, js가 불필요하다는 장점이 있습니다.

선택적 이미지 전송

모바일 퍼스트와는 달리, 서버 쪽에서 대상 스크린에 선택적으로 이미지를 보내는 접근방법으로, 대상 스크린의 크기를 취득하여 저장하는 구체적인 기법이 필요합니다. **User agent**를 이용한 방법도 가능하겠고, 아래와 같이 쿠키를 이용한 방법도 가능할 것 같습니다. 특히 **user agent**를 사용하는 방법은, 다음 강인 서버측 접근방법 (**RESS**)에서 더 자세히 다루겠습니다.

브라우저 쿠키를 이용한 선택적 이미지 화일 제공 기법

링크: [Responsive Images using Cookies](#)

웹페이지에서는 스크린 크기를 쿠키에 저장하고

```
document.cookie = "device_dimensions=" + screen.width + "x" + screen.height;
```

서버쪽에서 그 쿠키값을 이용하여 선택적으로 이미지 화일을 보냅니다. 이렇게 저장된 쿠키값을 이용하여 적절한 이미지화일을 전송하게 됩니다. (완전한 코드는 위 링크를 참조하세요)

```
<?php
    $device_width = 0;
    $file = $_SERVER['QUERY_STRING'];

    $dimensions = explode('x', $_COOKIE['device_dimensions']);
    $device_width = intval($dimensions[0]);

    $fileext = pathinfo($file, PATHINFO_EXTENSION);

    // Low resolution image
    if ($device_width <= 800) {
        $output_file = substr_replace($file, '-low', -strlen($fileext)-1, 0);
    }

    // check the file exists
    if (isset($output_file) && file_exists($output_file)) {
```

```

    $file = $output_file;
}

// return the file;
readfile($file);
?>

```

```

```

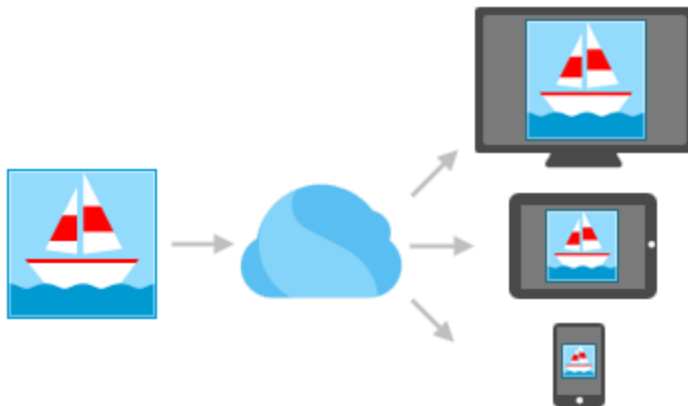
와 같이 사용하면, 화면폭이 작은 경우 (이 경우 800px), test.png 대신, test-low.png 를 전송하게 됩니다.

이러한 방법의 장점은, 모바일 퍼스트와는 달리, 한 이미지만 전송한다는 점입니다.
단점으로는, 쿠키를 사용해야 한다는 점입니다.

쿠키를 이용하는 방법으로 [Adaptive Image](#) 도 있습니다.

외부 서비스

대표적으로 [Sencha.io](#) 가 있습니다.



sencha의 서비스를 사용하게 되면, 자동으로 각 기기마다 최적으로 크기/품질이 조정된 이미지를 제공하는 서비스입니다.

이미지 다운로드 실험

비슷한 고민에서 시작된, 여러 브라우저에서 display:none 을 이용하여 선택적으로 다운로드 할 수 있는지의 실험이 있습니다.

원글: <http://timkadlec.com/2012/04/media-query-asset-downloading-results/>

제 블로그: <http://blog.gnuboard.org/2012/04/media-query-image-downloads/>

결론적으로, background-image 의 부모 태그에 display:none 을 적용하는 경우 (실험의 test # 3), 거의 모든 (모바일) 웹 브라우저에서 해당 백그라운드 이미지를 다운로드 하지 않는다고

합니다.

```
<div id="test3">
  <div></div>
</div>
```

```
#test3 div {
  background-image:url('images/test3.png');
  width:200px;
  height:75px;
}
@media all and (max-width: 600px) {
  #test3 {
    display:none;
  }
}
```

Tested	Requests Image
Android 2.1+	No
Blackberry (6.0+)	No
Chrome (16+)	No
Chrome Mobile	No
Fennec (10.0+)	Yes
Firefox (3.6+)	No
IE 9+	No
iOS (4.26+)	No
Kindle (3.0)	No
Opera (11.6+)	No
Opera Mini (6.5+)	No
Opera Mobile (11.5)	No
Safari (4+)	No

물론, 미디어쿼리로 각 기기별로 완전히 분리된 조건으로 다운로드 할 수도 있습니다.

```
<div id="test5"></div>
```

```

@media all and (min-width: 601px) {
    #test5 {
        background-image:url('images/test5-desktop.png');
        width:200px;
        height:75px;
    }
}
@media all and (max-width: 600px) {
    #test5 {
        background-image:url('images/test5-mobile.png');
        width:200px;
        height:75px;
    }
}

```

하지만 이렇게 되면, CSS3 미디어쿼리를 사용하지 못하는 경우, 백그라운드 이미지가 아예 사용되지 않는 문제점이 있습니다.

새로운 반응형 이미지 (표준)

이와 같이 별도의 기법으로 처리하지 말고, HTML 표준 (picture과 srcset)으로 넣으려는 노력이 있습니다.

반응형 이미지 그룹 (<http://responsiveimages.org/>)

- [반응형 이미지 W3그룹](#)
- [Picture Element 제안서](#)
- [HTML Responsive Images Extension 제안서](#)

picture 요소의 사용방법 예는 다음과 같습니다.

```

<picture>
  <source media="(min-width: 40em)" srcset="big.jpg">
  <source srcset="small.jpg 1x">
  
</picture>

```

예에서 보듯이, 미디어 쿼리가 source 속성으로 들어가서 해당 조건에 맞는 경우에 srcset으로 설정된 이미지 파일을 사용합니다. 여기서 사용된 srcset은 기존의 img 요소에도 적용될 수 있도록 제안되었습니다.

```



```

즉, 스크린 폭이 640px보다 작을 때는 small.jpg, 그렇지 않을 때는 med.jpg를 소스로 사용하고, 또한 고해상도 (2x)일 때는 각각 small-hd.jpg와 med-hd.jpg를 사용하도록 설정할 수 있습니다.

또한, 화일 크기 뿐만 아니라 (resolution switching), 화면 폭에 따라서 이미지의 어느 부분을 사용하는 가 (art direction) 도 고려되고 있습니다. 아래 그림에서 보듯이, 스크린의 크기에 따라서 이미지의 특정 부분만을 사용하도록 설정하는 접근방법입니다.



출처: responsiveimages.org use cases - <http://responsiveimagescg.github.com/ri-usecases/>

구현

위와 같은 표준을 현재 기술로 (비슷하게) 구현한 기법도 존재합니다.

picturefill.js

fillament group의 [Scott Jehl](#)가 [picturefill.js](#)를 이용하여 제안하는 방법입니다.

```
<div data-picture data-alt="A giant stone face at The Bayon temple in Angkor Thom, Cambodia">
  <div data-src="small.jpg"></div>
  <div data-src="medium.jpg" data-media="(min-width: 400px)"></div>
  <div data-src="large.jpg" data-media="(min-width: 800px)"></div>
  <div data-src="extralarge.jpg" data-media="(min-width: 1000px)"></div>
  <!-- Fallback content for non-JS browsers. Same img src as the initial,
```

```
unqualified source element. -->
<noscript>
  
</noscript>
</div>
```

`data-picture` 로 선언된 `div`에 대해서는 자동으로 ‘반응형 이미지’를 처리해줍니다.

jquery-picture.js

위 표준으로 제안된 태그들 (`picture/source`)을 직접 이용할 수 있도록 [jQuery 플러그인](#)이 제안되기도 했습니다.

```
<picture alt="A Half Brained Idea">
  <source src="assets/images/small.png">
  <source src="assets/images/medium.png" media="(min-width:440px)">
  <source src="assets/images/large.png" media="(min-width:600px)">
  <noscript>
    
  </noscript>
</picture>
```

```
$(function(){
  $('picture').picture();
});
```

HiSRC

다른 종류의 반응형 웹 이미지 `jquery` 플러그인입니다. <https://github.com/teleject/hisrc>
 다른 구현들이 미디어쿼리를 이용하여 화면폭에 따라서 이미지 소스를 결정한다면, 이 플러그인은 인터넷 연결속도와 디스플레이 해상도를 이용하여 보여줄 이미지를 결정합니다.

```
<div class="hisrc">
  
</div>
```

```
$(".hisrc img").hisrc();
```

위 코드는 우선 모바일퍼스트로, **200x100** 크기의 이미지를 로딩하고
 인터넷 연결속도를 체크하여 일정정도 이상 (기본 **300**) 이면 **1x** 짜리 (**400x200**) 이미지를,

그리고 해상도가 2x 이면 (예: 레티나) 800x400 크기의 이미지를 로딩합니다.

실제 속도체크는 <https://github.com/adamdbradley/foresight.js> 를 사용합니다.

Art direction by (pure) CSS

위 백안관앞 강아지 사진으로 예를 들었던 **art direction**를 구현한 **CSS** 입니다. 즉, (모바일에서) 크기가 줄어들어 이미지가 너무 작아질때, 특정 영역 (포칼포인트)을 중심으로 이미지의 일부만 보여주는 기법입니다.

설명이 부족하지만 원본글과 Git 링크입니다.

- <http://www.cdnconnect.com/docs/focal-point-css/pure-html-css-responsive-high-resolution-images-solution>
- <https://github.com/adamdbradley/focal-point>

사용방법은

```
<div class="focal-point right-3 up-3">  
  <div></div>  
</div>
```

처럼 이미지를 감싸는 **div**를 두개 만들고, 맨 상위 **div**에 **focal-point** 클래스 선언과 함께, 그림에서 어느 부분을 중심으로 할지를 **right-3 up-3** 식으로 알려주는 것입니다.

아래그림을 참조하세요. 출처는



출처: <http://designshack.net/articles/css/focal-point-intelligent-cropping-of-responsive-images/>

현재는 767px을 경계치로 사용하고 있어서 그 이하가 되면, 위에서 선택된 중심영역을 중심으로 이미지를 일부만 보여줍니다.



ORIGINAL



CROPPED

출처: <http://designshack.net/articles/css/focal-point-intelligent-cropping-of-responsive-images/>

CSS 구현방법을 살펴보면, 화면 폭이 기본 경계치로 잡혀있는 767px 이하가 되면, 12x12 셀좌표로 (가상으로) 나누어진 이미지를, 선택된 포칼포인트를 중심으로 이동하는 방식입니다.

```
@media all and (max-width: 767px) {

    /* 4x3 Landscape Shape (Default) */
    .focal-point div {
        margin: -3em -4em;
    }

    /* Landscape up (Total 6em) */
    .up-3 div {
        margin-top: -1.5em;
        margin-bottom: -4.5em;
    }

    .right-3 div {
        margin-left: -6em;
        margin-right: -2em;
    }
}
```

예를 들어 위에서 사용한 up-3 클래스의 경우, 각각 margin-top을 -1.5em, margin-bottom을 -4.5em 으로 조정하여

와 같이 이미지를 감싸는 div의 마진을 기본 폰트크기를 단위로 해서 조정하는 방법을 사용합니다..

폰트크기를 기본으로 하였기에, 767 이하의 화면 폭의 경우

```
@media all and (max-width: 550px) {
    h1 {
        font-size: 23px;
    }
}
```

와 같은식으로 폰트 크기를 지정하면 이미지의 크롬비율도 같이 조정되는 방식입니다.

기타 반응형 비 텍스트 요소들

그밖에 이미지는 아니지만, 비 텍스트 요소들의 반응형 기법들에 대한 (완전하지 않은) 자료 링크들입니다. 특히 반응형 광고는, 나중에 시간이 되면 더 많은 자료 조사 후 별도의 강의로 만들려고 합니다.

반응형 맵

- <http://bradfrostweb.com/blog/post/adaptive-maps/>

반응형 광고

- [Responsive Advertising](#)
- [Where Does Advertising Fit In A Responsive World?](#)
- [Adaptive Web-Design & Advertising](#)
- [Link Dump on Responsive Ads](#)

반응형 슬라이더

- <http://jasonweaver.name/lab/conditionalslideshow/>
- <http://responsive-slides.viljamis.com/>
- <http://webdesignledger.com/resources/8-responsive-jquery-slider-plugins> 자료모음

참고문헌/자료

- Ethan Marcotte, [Fluid Images](#), unstoppablerobotninja.com, 2009
- Chris Coyier, [Rundown of Handling Flexible Media](#), css-tricks.com, 2012
- [Fitvids.js](#)
- 전진, [반응형 이미지](#), 그누보드 배우기, 2012
- Tim Kadlec, [Media Query & Asset Downloading Results](#), 2012
- 전진, [미디어 쿼리와 이미지 다운로드](#), 그누보드 배우기, 2012
- [Responsive Images Community Group](#)
- [picturefill.js](#)
- [jQuery Picture](#)