

구글 액셀러레이티드 모바일 페이지(AMP) 적용하기

[구글 검색 콘솔에 액셀러레이티드 모바일 페이지\(AMP\)](#)라는 신기한 녀석이 출현했습니다. AMP는 모바일 환경에서 빠르게 로드되도록 설계된 페이지입니다. 자신의 웹사이트에 이를 적용하면 구글에서 제공하는 캐시에 저장되어 구글을 비롯한 여러 메타사이트가 웹페이지 프리뷰 등의 목적으로 활용할 수 있게 됩니다. 이전 글에 [간단한 소개](#)가 있으니 참고하세요.

[AMP HTML 사양](#)에 대한 기본적인 내용을 번역하기보다는 제가 이 작업을 수행하면서 경험한 삽질이나 알아낸 것, 유용하다고 생각되는 방법을 중심으로 설명하겠습니다. AMP 페이지를 작업하기에 앞서 선행되어야 할 사항으로 [구글 페이지 인사이트](#)에서 모바일 사용환경 테스트 점수를 80점 이상으로 통과한 상태여야 하며, 추가로 [데이터 구조화](#) 유형 중 아티클을 만족해야 합니다. 아티클 유형으로는 [Article](#), [NewsArticle](#), [BlogPosting](#) 또는 [VideoObject](#)입니다. 이 네 가지 유형에 속하지 않는 웹사이트는 AMP를 적용하는 것이 현재 무의미합니다. 그러면, 일반 게시판이나 위키, 쇼핑몰은 어디에 속할까요? 대부분 Article 유형에 속합니다.

Note: 선택적으로는 SSL을 지원하는 것이 좋습니다. `<amp-form>` 이나 `<amp-video>` 등 `<amp-img>` 를 제외한 URL 속성을 필요로 하는 대부분의 AMP 컴포넌트가 SSL 주소를 요구하기 때문입니다.

여기부터 작성되는 내용은 위 두 조건을 모두 만족한 것을 전제로 합니다. 이제 **AMP HTML을 작성**해 봅시다. 기존 HTML에 AMP 모드를 추가할 것인지 아니면 AMP 페이지를 별도로 제공할 것인지를 우선 결정해야 합니다. 여기에는 장점과 단점이 존재합니다. 전자는 작업효율이 높습니다. 이미 잘 돌아가고 있는 HTML을 건드릴 필요가 없으니까요. 스펙 문서를 보고 마구잡이로 때려 고쳐 내려가면서 검사기를 통과시키면 됩니다. 단점은 관리 포인트가 늘어나는 것이죠. 후자는 고스란히 그 반대입니다. 굉장히 하드코어한 코딩을 구사하게 될 거예요. 저는 이 두 가지 방법을 모두 시도하는 것을 추천합니다. 최초 페이지를 별도로 작업하다 보면 두 문서 간의 차이점을 쉽게 찾아낼 수 있고요. 결국에는 하나의 작업물에서 두 가지 아웃풋을 내는 형태로 병합할 수 있습니다.

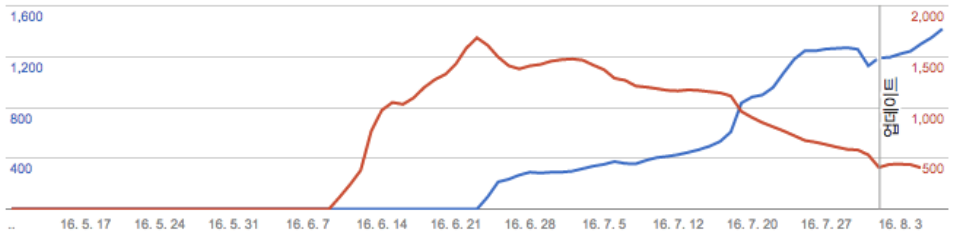
자, 이제 AMP HTML 스펙 문서를 열고 코딩하세요. 코딩하면서 아래의 유의해야 할 항목을 살펴보세요. 이것은 제가 범한 실수이고 오류가 보고될 때마다 기록한 것입니다. 작게 여겨질 수도 있겠지만, 오류 하나가 보고되면 이 오류 항목이 사라지기까지는 보통 1달 정도 혹은 그 이상의 시간이 소요된다는 점 꼭 유념하시기 바랍니다.

- AMP 구분자와 언어코드를 HTML에 삽입해 주세요. `<html amp lang="ko">`
- UTF-8이 기본입니다. `<meta charset="utf-8">`
- `<link rel="canonical">` 태그의 값에는 원래 문서의 고유 주소를 사용하세요.
- `<link rel="canonical">` 주소에는 필터와 같은 파라미터가 기입되지 않도록 하여 하나의 문서가 여러 링크를 만들어 내는 일이 없도록 해야합니다.

- m.firejune.com과 같이 모바일용 페이지를 별도로 제공하는 경우 `<link rel="alternate">` 를 이용하세요.
- `<meta name="viewport" content="width=device-width ...">` 필수 요소입니다.
- `<script async src="https://cdn.ampproject.org/v0.js">` 역시 필수 요소입니다.
- 위 스크립트가 삽입되면 주소창의 마지막에 `#development=1` 를 붙여 콘솔 로그를 통해 실시간으로 오류를 확인할 수 있습니다.
- 페이지에서 필요로 하는 스타일시트를 모두 `<style amp-custom>` 요소에 인라인으로 삽입하세요. 단, 50k를 넘겨선 안 됩니다.
- `<style amp-custom>` 에서 이미 정의된 `<amp-*>` 요소의 스타일을 재정의 해선 안 됩니다.
- `<style amp-custom>` 에서 CSS의 값에 `!important` 를 사용할 수 없습니다.
- `<head>` 에 선언된 `<style amp-custom>` 외 그 어떤 곳에도 `<style>` 태그는 허용되지 않습니다.
- 일반 `<script>` 태그도 마찬가지로 절대 허용하지 않습니다.
- 서드파티 라이브러리 포함하여 페이지에 삽입된 모든 자바스크립트를 삭제하세요.
- 요소에 인라인으로 작성된 `on*=` 이벤트와 `style=` 속성을 모두 삭제하세요.
- 별도의 확장 컴포넌트 없이 사용할 수 있는 `<amp-img>`, `<amp-video>`, `<amp-audio>`, `<amp-pixel>`, `<amp-ad>` AMP 태그를 치환하는 작업을 먼저 수행 하세요.
- 모든 `<amp-*>` 요소에는 `style=` 과 `on*=` 속성을 사용할 수 없습니다.

- `<amp-video>` 요소의 `src` 속성은 절대 경로여야 하며 `https` 프로토콜을 이용해야 합니다.
- **확장 컴포넌트**(추가적인 로딩이 필요한)에 상응하는 AMP 요소로 치환하고 동적으로 컴포넌트를 로딩하는 루틴을 구현하세요.
- 기존 `<iframe>` 으로 삽입된 youtube나 vimeo 동영상은 각 상응하는 컴포넌트를 이용하여 삽입해야 합니다.
`<amp-iframe>` 으로만 치환하면 서비스 종류에 따라서 오류가 발생하기도 합니다.
- `<form>` 의 자식 요소로 사용되는 모든 요소(예: `input`)는 `<form>` 안에서만 사용할 수 있습니다.
- 대부분의 AMP 디스플레이 요소에 사용할 수 있는 `layout` 속성의 값으로 "responsive"를 이용하면 크기를 자동으로 계산해 줍니다.
- AMP 디스플레이 요소의 필수 값인 `height` 와 `width` 속성의 단위는 픽셀이어야 합니다.
- `<amp-sidebar>` 요소는 `<body>` 바로 하위에 위치해야 합니다.
- 확장 컴포넌트는 다양한 방법으로 응용할 수 있습니다.
- 어느 정도 마무리되면 **웹 기반 유효성 검사기**를 돌려봅니다.
- 끝으로 `<link rel="amphtml">` 태그를 고유 문서의 `<head>` 에 삽입하고 크롤링 당합니다.
- `<link rel="amphtml">` 태그의 값에는 원래 문서 주소의 규칙을 유지하는 것이 좋습니다.
- 서로 다른 고유주소를 가진 문서가 하나의 `<link rel="amphtml">` 주소를 가르키지 않도록 해야합니다. 오류가 두 달 이상 사라지지 않고 있네요.
- `<amp-pixel>` 을 이용하여 캐시에 저장된 페이지로 사용자가 접근하는 것을 모니터링 하거나 할 수 있습니다.

- `<amp-iframe>` 에 삽입된 콘텐츠가 스크롤을 가지는 경우 나타나지 않을 수 있으며, 나타나지 않을 경우를 대비한 대체(placeholder) 이미지를 필요로 합니다.
- `<amp-form>` 을 이용하여 검색이나 댓글 작성기능을 추가할 수 있습니다. 단, SSL이어야 하며, CORS 이슈까지 처리해야 정상적으로 작동합니다.



Google Search Console(firejune.com) > Search Appearance > Accelerated Mobile Pages

모든 유효성 검사를 통과하고 크롤러가 문서를 굽어가면 **구글 AMP 캐시**에 저장되고 그 결과를 위 그림처럼 서치 콘솔에서 확인할 수 있습니다. 그리고 일정 시간이 지나면 이후 부터 **검색 결과에 노출**되기 시작합니다. AMP 캐시 서버에 저장되는 대상은 HTML 문서와 포함된 이미지 파일이며 그외 정적인 파일은 SSL기반의 핫링크가 걸리게 되죠. 캐시된 문서는 다음과 같이 REST API를 이용하여 접근 하거나, 갱신됨을 알리(핑)거나, 캐시된 URL 목록을 호출하거나, 삭제요청을 직접적으로 수행할 수도 있습니다.

```
# Request for an AMP HTML document
GET https://cdn.ampproject.org/c/s/example.com/amp_document.html
# Request for an image
GET https://cdn.ampproject.org/i/example.com/logo.png
# Request an AMP URL
POST https://acceleratedmobilepageurl.googleapis.com/v1/ampUrls:batchGet
```

```
# Update ping
GET https://cdn.ampproject.org/update-ping/c/s/ampbyexample.com
# Remove AMP content
GET https://cdn.ampproject.org/update-ping/i/s/example.com/favicon.ico
```

Note: 캐시 상태를 신속하게 동기화 할 수 있는 수단이며, 일반적으로 크롤러가 알아서 하기 때문에 선택사항입니다.

우와.... 이걸 다 언제 작업하나요.... 아.... 이 짓을 다시 한다는 것은 군대를 다시 들어가는 그런 느낌이군요. 그래서 조금이나마 도움을 드리고자 PHP로 작성한 Ampify 클래스를 공유합니다. 이 정적인 클래스는 [GeSHi](#) 라는 라이브러리를 이용하여 코드의 문법 강조 기능을 포함하고 있습니다. 응답이 많이 느려지지만 클라이언트에서 할 수 없으니 백-엔드에서 할 수밖에요. 나중에 확장 컴포넌트로 나오려나요? 어차피 기계가 방문할 페이지인데 좀 느려도 괜찮겠죠. 그리고 빠른 이미지 크기 추출을 위해 [fastimage.php](#)를 필요로 합니다. 간단한 사용법은 `Ampify::content($html)` 메서드를 호출하여 일반 HTML을 AMP로 변환합니다. 더 자세한 내용은 코드에 있습니다. ;)

```
/**
 * Ampify class
 *
 * @license MIT
 * @author Firejune
 * @version 0.4.19
 */
class Ampify {
    // An var of list for source code syntax highlighting
    private static $highlights = array();
    // An var of list for extnend component loading
    private static $components = array();
```

```

// Source code syntax highlight using GeSHi
private static function codeHelper($matches) {
    require_once('lib/geshi/geshi.php');
    $lang = str_replace('language-', '', $matches[1]);
    $code = rtrim($matches[2]);
    if ($lang == 'json' || $lang == 'jsx') $lang = 'javascript';
    if ($lang == 'html' || $lang == 'markup') $lang = 'html5';
    if ($lang == 'command') $lang = 'bash';
    if (!in_array($lang, self::$highlights)) {
        self::$highlights[] = $lang;
    }
    $geshi = new GeSHi(str_tag_on(strip_tags($code)), $lang);
    $geshi->enable_classes();
    $code = $geshi->parse_code();
    $code = preg_replace('/<[ ]*pre( [^>]*)?>/i', '<pre$1><code>', $code);
    $code = preg_replace('/</pre>/i', '</code></pre>', $code);
    return self::fixClassName($code);
}

// Return defined styles of syntax highlight
private static function getCodeStyle() {
    $geshi = new GeSHi;
    $languages = self::$highlights;
    foreach ($languages as $language) {
        $file = $geshi->language_path.$language.'.php';
        if (!file_exists($file)) {
            continue;
        }
        $geshi->set_language($language);
        $css .= preg_replace('/^\s*\s*\s*\s*\s*\s*/s', '',
$geshi->get_stylesheet(false));
    }
    return $css;
}

// Fix class names of syntax type
private static function fixClassName($code) {
    $entities = array('class="command"',
'class="html"', 'class="json"', 'class="jsx"');
    $replacements = array('class="bash"', 'class="html5"',
'class="javascript"', 'class="javascript"');
    return str_replace($entities, $replacements, $code);
}

```

```

}
// Retrun Accepted HTML5 tags and AMP components
private static function getAllowTags() {
    $html =
'<h1><h2><h3><h4><h5><h6><a><p><ul><ol><li><blockquote><q><cite><ins><del><st
rong>'

'<em><code><pre><svg><table><thead><tbody><tfoot><th><tr><td><dl><dt><dd><art
icle>'

'<section><header><footer><aside><figure><figcaption><time><abbr><div><span><
hr><br>'
    . '<kbd><u><b><i><s><small><caption><address><button><source>';
    $amp = '<amp-embed><amp-img><amp-pixel><amp-video><amp-audio><amp-
anim><amp-iframe><amp-fit-text>'
    . '<amp-access><amp-font><amp-slides><amp-ad><amp-list><amp-live-
list><amp-social-share>'
    . '<amp-lightbox><amp-carousel><amp-accordion><amp-youtube><amp-vimeo>';
    return $html.$amp;
}
// Ignoring not allowed attribute of AMP for all tags.
private static function ignoreHelper($m) {
    $tag = $m[0];
    $attributes = $m[2];
    preg_match_all(
'/(([\w-]+\s*(?:=\s*(?:("[^"]*"|'\'([^\']*')\'|(\w*[\s>]*)))?)?/usix',
$attributes, $matches);
    foreach ($matches[1] as $key => $val) {
        $attr = $matches[2][$key];
        if (!$attr) $attr = $matches[3][$key];
        if (!$attr) $attr = $matches[4][$key];
        if ($val == 'href' && strpos($attr, 'javascript:') !== false) {
            $tag = str_replace($attr, '#', $tag);
        }
        if (preg_match(
'/(on(click|mousedown|mouseup|mousemove|mouseout|mouseover|load)|style|summa
ry)/i', $val)) {
            $tag = str_replace($matches[0][$key], '', $tag);
        }
    }
}
}

```



```

    return $tag;
}
// An helper for image tag
private static function imageHelper($m) {
    $attr = $m[1];
    $args = array();
    preg_match_all(
        '/([\w-]+\s*(?:=\s*(?:"(\^")*"|\'(\^\')*\'|\s*(\^s>*))?)?/usix', $attr,
        $matches);
    # Allowed attr of <img>
    foreach ($matches[1] as $key => $val) {
        if (preg_match(
            '/(src|srcset|layout|heights?|alt|role|on|tabindex|placeholder|widths?|data-*
            |type|class)/i', $val)) {
            $attr = $matches[2][$key];
            if (!$attr) $attr = $matches[3][$key];
            if (!$attr) $attr = $matches[4][$key];
            $args[$val] = $attr;
        }
    }
    if (!$args['layout']) {
        $args['layout'] = 'responsive';
    }
    if (strpos($args['width'], '%') !== false) {
        $args['width'] = 0;
    }
    if (strpos($args['height'], '%') !== false) {
        $args['height'] = 0;
    }
    if (!$args['width'] || !$args['height']) {
        require_once('lib/fastimage.php');
        $src = $args['src'];
        if (preg_match('/\:\/\/(m\.)?firejune(\.cafe24)?\.com/i', $src)) {
            $src = preg_replace('/https?:\:\/\/(m\.)?firejune(\.cafe24)?\.com/i',
            '', $src);
        }
        $src = preg_replace('/^\.\.?\/\//', '/', $src);
        if (!preg_match('/(https?:)?\:\/\/[^\//]+\//i', $src) &&
        preg_match('/^\//i', $src)) {
            $src = '/public'.$src;
        }
    }
}

```

```

    }
    if (strpos($src, '/public/images/') === 0) {
        $args['layout'] = 'fixed';
    }
    $img = new FastImage('./.$src);
    $size = $img->getSize();
    if ($args['width']) {
        $args['height'] = intval($size[1] / $size[0] * $args['width'], 10);
    } elseif($args['height']) {
        $args['width'] = intval($size[0] / $size[1] * $args['height'], 10);
    } else {
        $args['width'] = $size[0];
        $args['height'] = $size[1];
    }
}
if ($args['width'] < 240 && $args['height'] < 240) {
    $args['layout'] = 'fixed';
}
if (!$args['src'] || !$args['width'] || !$args['height']) {
    return '';
}
$attr = '';
foreach ($args as $key => $val) {
    $attr .= ' '.$key.'="'.$val.'"';
}
return '<amp-img'.$attr.'></amp-img>';
}
// An helper for video tag
private static function videoHelper($m) {
    $tag = $m[0];
    $src = $m[1];
    $tag = preg_replace('/<amp-video(?:.*?)>/', '<amp-video$1
layout="responsive">', $tag);
    return str_replace($src, strip_tags($src, '<source>'), $tag);
}
// An helper for iframe tag
private static function iframeHelper($m) {
    $tag = $m[0];
    $src = $m[1];
    preg_match('/<iframe.*width="([\d+]"*.height="([\d+]"*.*/', $tag,

```

```

$m);
    $width = $m[1];
    $height = $m[2];
    # amp-youtube
    if (preg_match(
"/^(?:http(?:s)?:)?\:\/\/(?:www\.)?(?:m\.)?(?:youtu\.be\/|youtube\.com\/(?:?(w
atch)?\?(?:.*&)?v(?:i)?=|(?:embed|v|vi|user)\\/))([^\?&"]>)+)", $src, $id))
    {
        return '<amp-youtube data-videoid="'. $id[1].'" layout="responsive"
width="'. $width.'" height="'. $height.'"></amp-youtube>';
    }
    # amp-vimeo
    if (preg_match('/(https?:\:\/\/)?(www\.)?(player\.)?vimeo\.com\/([a-
z]*\/)*([\d0-9]{6,11})[?]?.*\/' , $src, $id)) {
        return '<amp-vimeo data-videoid="'. $id[5].'" layout="responsive"
width="'. $width.'" height="'. $height.'"></amp-vimeo>';
    }
    # amp-iframe
    return '<amp-iframe src="'. $src.'" width="'. $width.'"
height="'. $height.'" sandbox="allow-scripts" layout="responsive" class="bdr1"
frameborder="0"></amp-iframe>';
}
// Add using AMP extend component to list
public static function add($component) {
    if (is_array($component)) {
        $component = 'amp-'. $component[1];
    }
    if (!in_array($component, self::$components)) {
        self::$components[] = $component;
    }
}
// Convert contents to AMP components
public static function content($html) {
    $html = self::html($html);
    $html = self::html($html);
    # Remove unnecessary tags
    $html = preg_replace('/<(script|style).*?<\\/\1>\/s', '', $html);
    # Replace iframe tag to AMP component
    $html = preg_replace_callback(
'#<iframe.*?src="( [^" ]*" .*?[> ])*.*?<\/iframe>#si', 'ampify::iframeHelper',

```

```

$html);
    # Reduce ignore attributes
    $html = preg_replace_callback('/<([\w-]+) ([^>]*)?>/si',
'ampify::ignoreHelper', $html);
    # Code syntax highlight
    $html = preg_replace_callback(
'#<pre.*?class="([\^"]*)" .*?[\^]*><code>(.*?)</code></pre>#is',
'ampify::codeHelper', $html);
    # Whitelist of HTML tags allowed by AMP
    $html = strip_tags($html, self::getAllowTags());
    # Dynamic load amp components
    preg_replace_callback(
'</\amp-(youtube|vimeo|iframe|accordion|carousel)>/', 'ampify::add', $html);
    return $html;
}
// Convert generic HTML to AMP
public static function html($html) {
    # Replace img, audio, and video elements with amp custom elements
    $html = str_ireplace(
        array('<img', '<video', '/video>', '<audio', '/audio>'),
        array('<amp-img', '<amp-video', '/amp-video>', '<amp-audio', '/amp-
audio>'),
        $html
    );
    # Add amp attribute to html tag
    $html = preg_replace("/<[ ]*html( [^>]*)?>/i", '<html amp$1> ', $html);
    # Fix display amp custom elements
    $html = preg_replace_callback('/<amp-img(.*?)\/?>/',
'ampify::imageHelper', $html);
    $html = preg_replace_callback(
'#<amp-video.*?[\^]*>(.*?)</amp-video>#is', 'ampify::videoHelper', $html);
    return $html;
}
// Return stylesheets string from files
public static function css($path, $name) {
    $css = file_get_contents($path.$name.'/css/master.css');
    $css .= file_get_contents($path.$name.'/css/article.css');
    $css .= file_get_contents($path.$name.'/css/responsive.css');
    if (!empty(self::$highlights)) {
        $css .= self::getCodeStyle();
    }
}

```

```
}
    return $css;
}
// Return extended components when it using
public static function js() {
    foreach (self::$components as $component) {
        $js .= '<script async custom-element="'. $component. '"
src="https://cdn.ampproject.org/v0/'. $component. '-0.1.js"></script>';
    }
    return $js;
}
}
```

Note: 이곳의 사정에 맞게 코딩된 것으로, 모든 곳에서 작동한다는 보장은 못합니다.
참고용으로만 이용해 주세요.. :(

끝으로, 반가운 소식입니다. [AMP 프로젝트 페이지의 한글화](#) 작업이 진행
중이군요! 필독을 권장합니다. 그럼, 화이팅!