

Web Designing Web マスター
足立裕司の

リニューアル覚え書き

Web Designing 公式サイトをリニューアルした際のちょっとしたポイントを紹介

サイトも併せて見てください



text Web マスター
足立裕司 (9031) __ <http://9031.com/>

フリーランスの Web デザイナー。本誌 01 号 (2001 年) から現在まで Web Designing 公式サイトの Web マスターを担当。また本誌 2001 年から 2003 年までは「Web SPICES」を連載。初の iOS アプリ「Will Return」を 8 月にリリース。
<http://willreturn.at/>



Web Designing の公式サイトを、2012年7月18日に全面リニューアルしました。「今どきのサイトの標準的仕様+α」をコンセプトに、デザイン、使い勝手、コーディング、スマートフォン対応など、さまざまなアップデートを行っています。
<http://book.mynavi.jp/wd/>

Vol.4 マルチデバイス対応とレスポンシブ化の進行

Web Designing の公式サイトは、スマートフォン端末などでの表示にも最適化されました。最近のマルチデバイス対応は、レスポンシブ Web デザインの考え方が流行しています。徹底したものは、PC でのフルスクリーン表示からスマートフォンの画面サイズまで、リサイズすることにレイアウトがズルズルと変化するサイトもありますね。

そういった流行も横目に見つつ、Web Designing 公式サイトの場合は「半分くらいレスポンシブ」になりました。これまで通りの PC 向けのレイアウトと、転送データ量や表示するコンテンツをハンドヘルドなサイズに縮小・最適化したレイアウトの、大きくは 2 段階の表示切り替えとなっています。ウインドウ幅のブレイクポイントは 768 ピクセルとし、それ未満の場合にハンドヘルドなサイズの端末向けとして、フルードグリッド／フルードイメージなレイアウトに切り替わります。768 ピクセル以上の画面サイズでは、コンテンツの横幅が固定された PC 向けのサイトが表示されます。したがって iPad などのタブレット端末の画面サイズでは、PC 向けサイトが表示されます。

連載最後となる今回は、そういったレイアウトを切り替えるポイントを紹介していきます。なお、ソースはスペースの都合で短く抜粋してまとめていますので、実際のサイトの記述と多少異なることがあります。

スマートフォンサイズの端末では、表示する情報量やデータ転送量、JavaScript 部分の動作の重さなどの問題もありますが、前回ご紹介したような jQuery を使ったアニメーション／インタラクティブな部分のレスポンシブ化や、今回取り上げたマルチデバイス対応は、今後も随時最適化・アップデートをしていくと思いますので、ご期待ください。

CSS でメディアクエリの切替え

トップページのスマートフォン向け表示



メディアクエリは、link 要素の media 属性や CSS の @media 属性を利用し、ウインドウの幅によって適用するスタイルを変更します。本サイトの場合は、PC 向けの CSS をベースにして、スマートフォン向けの最適化を行いました。CSS は単独のファイルにまとめて、@media 規則で振り分けて記述しています。メディアクエリに対応していない IE 8 以前のブラウザのためには、css3-mediaqueries-js (<http://code.google.com/p/css3-mediaqueries-js/>) を使用することにより、対応させることができます

```
/* すべてのウインドウ幅に共通のスタイル */
.element { ... }
@media only screen and (min-width: 768px) {
  /* ウインドウ幅が768ピクセル以上のスタイル */
  .element { ... }
}
@media only screen and (max-width: 767px) {
  /* ウインドウ幅が768ピクセル未満のスタイル */
  .element { ... }
}
```

jQueryでViewportの書き換え

iPad横位置表示 (幅 1,024 ピクセル)



iPad縦位置表示 (幅 768 ピクセル)



```
//$ (document).ready (); より前に記述でも動作可能
if (screen.width >= 768) {
  // ウィンドウ幅が 768 ピクセル以上の端末では表示をコンテンツの幅 (1,024 ピクセル) に合わせ、拡大縮小可能とする
  $("head").append('<meta name="viewport" content="width=1024">');
} else {
  // ウィンドウ幅が 768 ピクセル未満の端末では表示を端末画面の幅に合わせ、拡大縮小不可とする
  $("head").append('<meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1, maximum-scale=1, user-scalable=no">');
}
```

スマートフォンやタブレットでは、ウィンドウ幅 768 ピクセル未満でスマートフォン向けレイアウトが表示され、768 ピクセル以上で PC 向けレイアウトが表示されるようになっています。しかし Viewport (PC におけるブラウザのウィンドウサイズに該当する数値) をスマートフォン向けレイアウトのために device-width (端末画面の横幅) に設定していると、タブレットなどウィンドウ幅 768 ピクセル以上の端末で、PC 向けレイアウトが横にはみ出して表示されてしまう恐れがあります。これを解消するために、Viewport の設定を jQuery で書き換え、幅 768 ピクセル以上の端末では、PC 向けレイアウトがコンテンツの幅にフィットして表示されるようにしています

jQueryで要素の表示・非表示の書き換え

PC向け表示の共通フッタ



スマートフォン向け表示の共通フッタ



```
$(function(){
  if (screen.width >= 768) { // ウィンドウ幅が 768 ピクセル以上なら
    $("#gfooter").load("footer-pc.html");
    // PC向けフッタ footer-pc.html を #gfooter に読み込む
  }
});
```

スマートフォン向けの表示では、マイナビグループ共通のフッタにあるリンク群やサイドバーのパナー群が非表示になります。これらは外部テンプレートファイル化され、jQueryのload()で読み込んで書き出しています。スマートフォン向けには、これを読み込まないようにするか、別のテンプレートファイルを読み込むことにより、内容を簡略化したり、転送データ量を抑えることができます。読み込むか否かは、ウィンドウの横幅で判別しています

jQueryでナビゲーションのアコーディオン化

PC向け表示



スマートフォン向け表示・展開前



スマートフォン向け表示・展開後



```
<nav id="gnav">
  <h2>Contents</h2>
  <div id="#gnav-content">
    <ul>
      <li><a href="...">ホーム</a></li>
      <li><a href="...">最新号</a></li>
      <li><a href="...">バックナンバー</a></li>
    </ul>
  </div>
</nav>
```

```
$("#gnav > h2").toggle(function(event){
  // トリガーをクリックしたら以下の動作を交互に実行
  $("#gnav-content").stop().slideDown("fast");
  // スライドダウンで表示
}, function(){
  $("#gnav-content").stop().slideUp("fast");
  // スライドアップで非表示
});
```

PC 向けの表示で横一列に並べたグローバルナビゲーションは、項目数が増えると、スマートフォン向けの表示で大きさやレイアウトに無理が生じる場合があります。本サイトのスマートフォン向けレイアウトでは、アコーディオン方式でまとめて表示するようにしました。開閉方法は前回ご紹介したアコーディオン表示と同じく、jQueryで行っています。トリガーとなる要素をクリック(タップ)すると指定の要素をスライドダウン、またはスライドアップで、表示/非表示をトグル的に切り替えます