Engineer

レスポンシブデザイン

表示する画面に合わせて変化するレイアウト

どのようなものか

レスポンシブデザインは、端末の種類ごとに別のサイトを作るのではなく、縦向きの細いスマートフォンから幅の広いモニターまで、表示する画面に合わせて一つのページのレイアウトを変える設計手法です。より長い呼び方であるレスポンシブウェブデザインは、その名前の由来となった記事の題名でもあります。Ethan Marcotteが2010年5月25日にA List Apartで公開しました。発想を得たのは、彼が新たな分野と紹介したレスポンシブ建築です。人が近づくと、壁や設備が動いたり形を変えたりします。

Marcotteは、流動的なグリッド、柔軟な画像、メディアクエリーを「レスポンシブウェブデザインの三つの技術的要素」と呼びました。流動的なグリッドは、列の幅を固定ピクセルではなく、利用できる空間に対する割合で決めるため、ウィンドウとともに伸び縮みします。柔軟な画像は、それを収める列より広くならないようにします。メディアクエリーは、割合だけでは成り立たなくなる地点で、スタイルシートを切り替える手段です。たとえば、3列では細すぎて読めなくなったら、ページを1列にします。

2026年2月19日付のW3C勧告候補草案であるMedia Queries Level 4仕様は、メディアクエリーを、真か偽の二つの値を取る論理式と定義しています。文書を表示するブラウザーや端末の何らかの側面を判定する式です。レスポンシブなレイアウトで中心になるのは `width` で、画面ではスクロールバーを含むビューポートの幅を意味します。クエリー内の長さも、ほかと同じCSSの長さです。そのため、そこでのピクセルは物理ピクセルではなくCSSピクセルであり、 `em` は、読者が変更できるブラウザーの初期フォントサイズを基準にし、ページ自身が設定したサイズは基準にしません。

ページ側が指定しなければ、スマートフォンではこの仕組みが機能しません。ビューポートのmeta要素がない場合、モバイルブラウザーは、通常980から1024CSSピクセル幅のデスクトップウィンドウであるかのようにページを配置し、結

果を縮小します。どのメディアクエリーにもデスクトップの幅が見えるわけです。CSS Viewportの草案は、これを大きな画面だけを想定したページへの救済策で、小さな画面でも適切に動くページには不要と説明しています。後の例の `width=device-width, initial-scale=1` は、このページが後者であると伝える指定です。

概念を判断に生かす

「レスポンシブデザインとは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 読み込みが速く、スマートフォンでも使え、お客様がどのブラウザーを使っていても同じように動く画面を提供します。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 フルスタックのGIS地図開発を指揮 — DBからブラウザーまでのフルスタック地図開発

フルスタックのGIS地図開発を指揮 お問い合わせ →

なぜ重要なのか

Marcotteの主張は、見た目だけでなく保守にも同じだけ関わります。増え続ける端末に、互いに切り離されたデザインを合わせるのではなく、同じ体験の異なる側面として扱うことを提案しました。ページ、アドレス、スタイルシートを一つずつにすれば、一度の修正がすべての読者に届きます。そもそも特定の画面サイズ向けに書かれていないため、新たな画面サイズにも同じサイトで対応できます。

多くのチームが見落としがちなのは、ズームのためにも必要だという点です。拡大するとCSSピクセルが大きくなり、CSSピクセルで数えるビューポートの幅は狭くなります。2024年12月12日付のW3C勧告であるWeb Content Accessibility Guidelines 2.2は、レベルAAの達成基準1.4.10「リフロー」で、利用できる幅が320 CSSピクセルのときに、二方向のスクロールなしでコンテンツを利用できることを求めています。1,280 CSSピクセル幅のウィンドウを400%に拡大した状態に相当することも明記しています。その初期幅では、ノートパソコンのブラウザーを400%に拡大する読者を、ページのすべてのメディアクエリーは320ピクセル幅のビューポートを使う読者として扱います。W3Cの解説は、サイトが小さなビューポートに合わせるため、一般にレスポンシブデザインを使うことに触れています。小さなビューポートに収まるコンテンツは、スマートフォンの利用者だけでなく、大きな端末で拡大する人にも役立ちます。

三つ目の理由は転送量です。HTML Standardは、一つの画像に複数のファイルを用意する理由を三つ挙げています。ビューポートに応じた選択、見せたい構図に応じた選択、画像形式に応じた選択です。同じ写真でも、スマートフォンの幅い

っぱいに表示するときは、大きなモニターの幅いっぱいに表示するときより、はるかに少ないピクセルで足ります。`srcset` と `sizes` を使えば、スクリプトなしでブラウザーが小さなファイルを取得できます。

ページが応じられるものは、幅だけではありません。Media Queries Level 4は、主なポインティングデバイスも記述しています。`(pointer: coarse)` は、タッチスクリーンなど精度の低いポインティングデバイスに一致し、`(hover: none)` は、ホバーできない主なポインティングデバイスか、ポインティングデバイス自体がない場合に一致します。Marcotteの記事も、小さな画面でリンクの操作対象を大きくするために、メディアクエリーを使うことをすでに提案していました。

実際の活用例

スマートフォンでは1列、横向きのタブレットでは2列、広い画面では3列になるカード一覧を作るには、まずビューポートの指定と、ブラウザーが自ら選べる画像を用意します。

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

<img
  src="pool-400.jpg"
  srcset="pool-400.jpg 400w, pool-800.jpg 800w, pool-1600.jpg 1600w"
  sizes="(max-width: 40em) 100vw, (max-width: 64em) 50vw, 33vw"
  width="1600"
  height="900"
  alt="A tide pool at low water"
/>
```

レイアウトはスタイルシートで定めます。

```
.cards {
  display: grid;
  grid-template-columns: 1fr;
  gap: 1rem;
}

@media (width >= 40em) {
  .cards {
    grid-template-columns: repeat(2, 1fr);
  }
}

@media (width >= 64em) {
```

```
  .cards {
    grid-template-columns: repeat(3, 1fr);
  }
}

.cards img {
  max-width: 100%;
  height: auto;
}
```

流動的なのはグリッドです。`1fr`の列が、そのとき利用できる幅を分け合うので、固定サイズはありません。二つのメディアクエリーが切り替えを担い、`em`で書かれているため、読者の既定の文字サイズに従います。既定のフォントを大きくした人には、ほかの人ならすでに2列になるウィンドウ幅でも、1列で表示されます。その文字サイズでは一行に必要な幅が増えるため、適切な動作です。最後の規則は、すべての画像を列の中に収めます。

`sizes`属性は、まだ何も配置していない段階で、画像をどの幅で表示するかをブラウザーに伝えます。最初に真になった条件が採用され、どれも真でなければ最後の値を使います。ビューポートが39em幅なら最初の条件が真なので、画像は39em幅として扱われます。41em幅なら二つ目の条件が真になり、画像はビューポートの半分である20.5em幅として扱われます。ウィンドウが少し広い方で、小さいファイルが必要になるのは、レイアウトが変わったためです。HTML Standard自身の例も、この現象を順に説明しています。ブラウザーは、これらの数値と、`w`記述子を並べた`srcset`から、その画面上で各ファイルがどの画素密度になるかを求めます。そして画面のピクセル密度、ズーム率、場合によっては通信状況も考慮し、どのファイルでも選べます。`width`属性と`height`属性は、ファイルのダウンロード前に画像の場所を確保できるようにします。

このサイトのブラウザー検査を速くした実績では、同じ発想を検査に生かしています。2026年8月に行ったこの作業で、ヘッドレスブラウザーを動かす検査に、スタイルシートに実際にあるすべてのメディアクエリーから導いた、22種類のウィンドウ寸法をまとめた共通の表を渡しました。その結果、デザインが想定するすべての寸法でページを調べるレイアウト検査の所要時間は、95秒から405秒になりました。寸法は端末の一覧ではなくスタイルシートから得るため、ページは、デザインに実在するブレークポイントで検査されます。

問題が起きるところ

端末の寸法を調べるブレークポイント。 Marcotteの2010年の例は、画面全体の幅である`max-device-width`を調べていました。Media Queries Level 4は現在、`device-width`、`device-height`、`device-aspect-ratio`を非推奨とし、作成者は使

ってはならないと定めています。モバイル端末かどうかを判別する手掛かりとして使われがちですが、ページが実際に使える空間については何も示さないとも説明しています。代わりに `width` を調べ、人気のスマートフォンの幅ではなく、コンテンツが成り立たなくなる地点に各ブレークポイントを置きます。

二つのブレークポイントの隙間。 一方のレイアウトに `(max-width: 320px)`、次に `(min-width: 321px)` を使えば、すべてを覆うように見えます。しかし仕様が指摘するように、320ピクセルより大きく321ピクセルより小さい幅のビューポートは、どちらにも一致しません。整数でないピクセル密度の画面やズームによって、小数の幅は実際に生じます。その幅では、どちらのスタイル群も適用されません。 `(width <= 320px)` と `(width > 320px)` という範囲構文で、どの幅も必ず一方だけに入るようにします。

ズームを無効にする。 ビューポートの指定に `user-scalable=no` があるか、`maximum-scale` が2未満だと、文字を200%に拡大できることを求める達成基準1.4.4「テキストのサイズ変更」のW3Cテスト規則を満たしません。同規則は、デスクトップブラウザーはこの要素を参照せず、モバイルでも現在の大半のブラウザーは初期状態から指定を無視するか、指定に関係なく拡大できるアクセシビリティー設定を用意していると説明しています。そのため、なお影響を受けるのは古いモバイルブラウザーの利用者です。両方のプロパティーを省き、代わりに、拡大しても使えるレイアウトにします。幅の狭いブレークポイントでフォントを小さくする場合も、同じ注意が必要です。W3Cのリフローの解説は、それを認めていますが、何らかの方法で通常の2倍の文字サイズにできることが条件です。

レイアウトを無視する画像。 `w` 記述子を `srcset` で使い、 `sizes` を指定しないと、HTML Standardに従うブラウザーは、画像がビューポート全体を埋めると仮定します。そのため、画面の3分の1に表示する画像でも、必要な約3倍の幅のファイルを取得することがあります。 `sizes` ではパーセントは使えず、 `vw` を使います。また、 `srcset` が `w` 記述子を使う場合、 `srcset` を理解するブラウザーは `src` を無視するので、 `src` だけを変えても、大半の読者には何も変わりません。 `sizes` をCSSに合わせ、レイアウトを変えるときは両方を変更します。スクリプトで画像を選んでも解決しません。標準は、画像を二重にダウンロードする場合や、スクリプトを無効にした読者にはまったくダウンロードしない場合があると警告しています。

一つの幅広い要素がページを横に押し広げる。 データ表、長いコード行、幅を固定した埋め込みは320 CSSピクセルを超え、ページ全体に横スクロールを生じさせることがあります。その結果、どの段落も左右に往復しながら読むことになります。W3Cのリフローの解説は、データ表、地図、動画など、二次元が必要な

コンテンツを例外としますが、その周囲の段落は例外ではありません。表を `overflow-x: auto` で独立してスクロールする専用のコンテナーに入れ、画像と動画には `max-width: 100%` を指定して、必要な部分だけがスクロールするようにします。

狭い画面をタッチスクリーンと見なす。 ノートパソコンを400%に拡大すれば表示幅は狭くなりますが、マウスがあります。大きなタブレットは幅が広くても、ホバーはまったくできません。レイアウトは `width` 、操作対象の大きさとホバー効果は `pointer` と `hover` で判断します。一つの端末に複数のポインティング手段があり得るため、Media Queries Level 4は、 `any-pointer` と `any-hover` も考慮するよう勧めています。

参考資料

- レスポンシブウェブデザイン — 手法と三つの要素に名前を付けた、2010年5月25日のEthan Marcotteの記事です。
- Media Queries Level 4 — `width` 特性、範囲構文、相対単位、ポインターとホバー、および `device-width` が非推奨となった理由を説明します。
- HTML Standard：画像 — `srcset` 、 `sizes` 、複数ファイルを用意する三つの理由、29emと32emを使った具体例です。
- 達成基準1.4.10「リフロー」の解説 — 320 CSSピクセルとする理由、ズームとの関係、例外となるコンテンツを説明します。
- ビューポートのmeta要素がズームを許可する — `user-scalable` と `maximum-scale` のW3Cテスト規則と、ブラウザーの対応状況に関する注記です。
- CSS Viewport Module Level 1 — ビューポートのmeta要素、そのプロパティー、モバイルブラウザーがデスクトップ幅を仮定する理由を説明します。

用語集

実際の制約に合わせて選ぶ

業務の流れ、既存のインターフェース、変更を必要とする利用者をお知らせください。検証可能な納品内容と、お客様のチームが保守を続けられる引き継ぎ内容を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「レスポンシブデザインとは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/responsive-design/