Engineer

ビューポート

ウィンドウが実際に表示するページの範囲

どのようなものか

ビューポートとは、ブラウザーが文書を配置して描画する長方形の領域です。画面ではウィンドウ自体のツールバーを除いた内側、紙では各用紙のページ領域に当たります。CSSは常にこれを基準に寸法を決めます。ルート要素を配置するボックスである初期包含ブロックはビューポートと同じ大きさになり、`width` や `height` のメディアクエリーはその寸法を調べ、`vw` と `vh` はその100分の1を表します。スマートフォンではviewportや複数形のviewportsだけでは区別できず、仕様はレイアウト、視覚、小、大、動的という修飾を設けています。

デスクトップでは、ビューポートとウィンドウはほぼ同じものです。スマートフォンでは両者が分かれます。デスクトップのウィンドウ向けに設計された文書が多いため、モバイルブラウザーは指定のないページをデスクトップに似せた固定幅で配置し、その結果を画面に収まるよう縮小します。CSS Viewport仕様によれば、その幅は通常980〜1,024 CSSピクセルです。ページはhead内の1要素でこの扱いを変更できます。その要素は `width` 、 `height` 、3種類の拡大縮小制限、 `user-scalable` 、 `interactive-widget` を受け付けます。

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

ズームするとビューポートは2つに分かれます。レイアウトビューポートはCSSがページ配置の基準にする領域で、ピンチズームでは変わらないため、改行位置も変わりません。視覚ビューポートは、その時点で画面に見えているページの範囲です。拡大すると狭まり、パン操作をすると移動します。既定では画面キーボードが縮めるのは視覚ビューポートだけです。 `interactive-widget=resizes-content` はレイアウト側も変更するよう求め、 `overlays-content` はどちらも変更しないよう求めます。

スマートフォンには、表示領域から引っ込むブラウザーの操作部もあります。アドレスバーは読み込み時には場所を取り、読者がスクロールすると引っ込むため、ページ自体が動かなくても見える高さが変わります。CSS Values 4はこのために3つの大きさを定めています。大ビューポートは収納可能なツールバーがすべて引っ込んだ状態、小ビューポートはすべて表示された状態を想定し、動的ビューポートはその時点の状態に追従します。単位にはそれぞれ `lvh` 、 `svh` 、 `dvh` という接頭辞が付き、幅に対応する単位もあります。接頭辞のない `vh` と `vw` は大ビューポートを基準に定義されています。画面キーボードに多い、コンテンツを動かさずに重ねて表示するインターフェースは、これらの長さを変えません。

概念を判断に生かす

「ビューポートとは？」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

レスポンシブデザインはビューポートを土台にしています。スタイルシートの幅のブレークポイントは、その幅を調べる条件です。したがって `meta` 要素を省いたページは、メディアクエリーをどれほど丁寧に書いても、スマートフォン上でデスクトップ幅に配置され、文字が読めないほど縮小されます。

難しいのは高さです。幅は回転やリサイズで変わることがあります。高さもツールバーの出入り、キーボードの表示、端末の回転によって変わることがあります。そのため、画面の高さに合わせる要素は、どの高さを指すかを明確にしなければなりません。小ビューポートを基準にすれば、収納可能なツールバーが表示された状態で収まるようにコンテンツを配置でき、バーが引っ込むと空白の帯が残ります。大ビューポートなら画面を満たせますが、ツールバーが出ている間はその背後にコンテンツが隠れます。動的ビューポートはツールバーの状態変化を反映します。ただし仕様自体が、スクロール中にコンテンツの大きさを変え、その下のすべてを動かし得ると警告しています。

画面には角もあります。角が丸かったりカメラ用の切り欠きがあったりするスマートフォンは、長方形ではありません。既定の `auto` では、その画面上でもページ全体を見られるようにします。 `viewport-fit=cover` を加えると、物理画面全体

を囲む長方形をビューポートにするよう求めることになり、`env(safe-area-inset-*)` 変数を使って文字を切り欠きから離す責任はページ側に移ります。長方形のディスプレイではこの余白値がゼロでなければならないため、同じパディング規則で両方に対応できます。

固定配置もビューポートを基準にします。固定ボックスはレイアウトビューポートに対して配置され、レイアウトビューポートの寸法はCSS Positioned Layoutで動的ビューポートに一致するよう定められています。そのため、Cookieバー、モーダルの背景、全画面メニューはページをスクロールしても動きません。ただし、祖先要素がその基準を引き受けていないことが前提です。`transform` 、`will-change` 、 `contain` はその基準になり得ます。

印刷は、画面専用のスタイルシートが想定していないビューポートです。紙ではビューポート単位がページ領域を測り、 `width` メディアクエリーはページボックスの幅を調べ、固定要素はすべてのページに繰り返し現れます。

実際の活用例

このサイトのページは `width=device-width, initial-scale=1, viewport-fit=cover` を宣言しています。したがってスマートフォンの幅で配置され、切り欠きの下も含めて画面の端まで広がります。後半の指定への対応はスタイルシートが担います。左右のパディングは `max(1rem, env(safe-area-inset-left))` とその右側版なので、文字は端から1rem離れ、切り欠きがある場所だけさらに内側へ移ります。高さの低い横向き画面では、最小の余白を `max(2rem, 7vw, env(safe-area-inset-left))` へ増やします。切り欠きのない端末ではインセットがゼロになり、1remでは端に近すぎたためです。

全画面セクションの高さは、 `100dvh` に設定した共有変数から取り、代替として `100vh` を使います。そばのコメントには理由も残っています。 `100vh` はアドレスバーが出ているときの可視領域より高くなることがあり、ヒーロー領域の下から次のセクションがのぞいていました。2つの要素は意図的にこの変数を使いません。マストヘッドのマークには静的な `vh` を使います。動的ビューポートに連動させると、ブラウザーの操作部が引っ込む際にマークの大きさが変わり、スクロールの途中でヒーロー領域が再配置されてしまうためです。タッチ画面の固定背景は高さを `100lvh` にして、アドレスバーの出入りで大きさが変わらないようにしています。また、独立した合成レイヤーに載せています。タッチ操作用ブラウザーが慣性スクロール中に背景を明暗の帯へ分断したためです。

以前のホームページのレイアウトでは、各サービスブロックの最小の高さを `100dvh` にしていたため、内容が画面より短い間は一画面を占めていました。2026年11月に確認したスタイルシートのコメントには、852×393ピクセルでのテストが記録されています。ブロックの内容は高さ412ピクセルに達し、隣のブロックと接していました。そのレイアウトでは、 `box-sizing: border-box` で `100dvh` の内側にパディングを含めると、背の高い画面では余分な高さを取らず、内容がはみ出す場合にだけ間隔が生まれました。

記録されている2026年8月の品質ゲート実装段階では、メディアクエリーに合わせてテストしていました。ブラウザーで動くこのサイトの品質ゲートは、スタイルシートのすべてのメディアクエリーから作った22種類のウィンドウ形状の共有表へ検査範囲を広げました。その結果、レイアウト検査だけで所要時間が95秒から405秒になりました。

印刷ではエンジンの違いが表面化しました。印刷用スタイルシートの記録では、あるエンジンはビューポート単位をページボックスに、別のエンジンはウィンドウに対して解決していました。そのため後者では、幅の広いウィンドウから印刷すると各行の3分の1が失われました。マストヘッドのスタイルシートには、同じ問題のメディアクエリー側の記録もあります。印刷時にもウィンドウの向き、高さ、幅が使われたため、広いウィンドウと狭いウィンドウから印刷した用紙ではマストヘッドに47ピクセルの差が出ました。クエリーの先頭に `screen and` を付け、規則を画面のウィンドウだけに限定して解決しました。

包含ブロックの落とし穴も記録されています。動きを減らす設定を実態に合わせた作業では、5つの異なるプロパティがそれぞれ、ビューポートに対して配置される要素の包含ブロックを祖先要素へ変えてしまうことが分かりました。それによって、閉じる操作を受けるレイヤーが知らないうちに画面の一部へ縮んでいました。

問題が起きるところ

スマートフォンに小さなデスクトップ用ページが表示される。 ビューポートの `meta` 要素がないと、ブラウザーはデスクトップ幅でページを配置してから縮小し、メディアクエリーもその幅を見ます。 `width=device-width, initial-scale=1` を追加し、ズームは有効なままにします。 `user-scalable=no` は、小さな文字を拡大する読者自身の手段を奪います。

全画面セクションが画面の下端を越える。 `100vh` は大ビューポートなので、アドレスバーが出ている間はセクションの下端も、そこに置いたボタンもバーの背後に隠れます。展開されたブラウザーのツールバーを避けて収めるには `svh`

を使い、`dvh` はスクロール途中の大きさの変化を許容できる場所だけに使います。

全画面のオーバーレイがページ内の小さな箱になる。 `transform`、それを指定する `will-change`、または一部の包含設定を持つ祖先要素は、`position: fixed` の包含ブロックになることがあります。その場合、オーバーレイはその祖先に合わせた大きさになります。変形を適用した要素の外側にオーバーレイを描画します。

印刷したページで各行の末尾が失われる。 ビューポート単位とウィンドウ寸法のメディアクエリーが印刷にも適用され、ウィンドウ基準で解決するエンジンが用紙を画面向けに配置してしまいます。印刷する部分にはビューポート単位を持ち込まず、ウィンドウ向けのクエリーには `screen and` を前置します。

参考資料

- CSS Viewport Module Level 1 — ビューポートの `meta` 要素と、`interactive-widget` の3つの動作。
- CSS Values and Units Level 4のビューポート百分率長 — 小、大、動的ビューポートの大きさと、それらを基準にするすべての単位。
- CSS Positioned Layout Level 3 — 固定配置、その基準を引き受けるプロパティ、紙への出力時の動作。
- CSS Environment Variables Level 1 — セーフエリアのインセットと、長方形の画面ではゼロにする規則。
- Media Queries Level 4 — 画面ではビューポート、印刷ではページボックスに対して測る `width`、`height`、`orientation`。

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

「ビューポートとは？」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/viewport/